List-Decodable Robust Mean Estimation and Learning Mixtures of Spherical Gaussians

Ilias Diakonikolas
University of Southern California
USA
diakonik@usc.edu

Daniel M. Kane
University of California, San Diego
USA
dakane@cs.ucsd.edu

Alistair Stewart
University of Southern California
USA
stewart.al@gmail.com

ABSTRACT

We study the problem of *list-decodable (robust) Gaussian mean estimation* and the related problem of *learning mixtures of separated spherical Gaussians*. In the former problem, we are given a set T of points in $\mathbb{R}^n$ with the promise that an α-fraction of points in T, where $0 < \alpha < 1/2$, are drawn from an unknown mean identity covariance Gaussian G, and no assumptions are made about the remaining points. The goal is to output a small list of candidate vectors with the guarantee that at least one of the candidates is close to the mean of G. In the latter problem, we are given samples from a k-mixture of spherical Gaussians on $\mathbb{R}^n$ and the goal is to estimate the unknown model parameters up to small accuracy. We develop a set of techniques that yield new efficient algorithms with significantly improved guarantees for these problems. Specifically, our main contributions are as follows:

List-Decodable Mean Estimation. Fix any $d \in \mathbb{Z}_+$ and $0 < \alpha < 1/2$. We design an algorithm with sample complexity $O_d(\mathrm{poly}(n^d/\alpha))$ and runtime $O_d(\mathrm{poly}(n/\alpha)^d)$ that outputs a list of $O(1/\alpha)$ many candidate vectors such that with high probability one of the candidates is within ℓ_2-distance $O_d(\alpha^{-1/(2d)})$ from the mean of G. The only previous algorithm for this problem achieved error $\tilde{O}(\alpha^{-1/2})$ under second moment conditions. For $d = O(1/\varepsilon)$, where $\varepsilon > 0$ is a constant, our algorithm runs in polynomial time and achieves error $O(\alpha^\varepsilon)$. For $d = \Theta(\log(1/\alpha))$, our algorithm runs in time $(n/\alpha)^{O(\log(1/\alpha))}$ and achieves error $O(\log^{3/2}(1/\alpha))$, almost matching the information-theoretically optimal bound of $\Theta(\log^{1/2}(1/\alpha))$ that we establish. We also give a Statistical Query (SQ) lower bound suggesting that the complexity of our algorithm is qualitatively close to best possible.

Learning Mixtures of Spherical Gaussians. We give a learning algorithm for mixtures of spherical Gaussians, with unknown spherical covariances, that succeeds under significantly weaker separation assumptions compared to prior work. For the prototypical case of a uniform k-mixture of identity covariance Gaussians we obtain the following: For any $\varepsilon > 0$, if the pairwise separation between the means is at least $\Omega(k^\varepsilon + \sqrt{\log(1/\delta)})$, our algorithm learns the unknown parameters within accuracy δ with sample complexity and running time $\mathrm{poly}(n, 1/\delta, (k/\varepsilon)^{1/\varepsilon})$. Moreover, our algorithm is robust to a small dimension-independent fraction of corrupted data. The previously best known polynomial time algorithm required separation at least $k^{1/4}\mathrm{polylog}(k/\delta)$. Finally, our algorithm works under separation of $\tilde{O}(\log^{3/2}(k) + \sqrt{\log(1/\delta)})$ with sample complexity and running time $\mathrm{poly}(n, 1/\delta, k^{\log k})$. This bound is close to the information-theoretically minimum separation of $\Omega(\sqrt{\log k})$.

Our main technical contribution is a new technique, using degree-d multivariate polynomials, to remove outliers from high-dimensional datasets where the majority of the points are corrupted.

CCS CONCEPTS

• **Theory of computation → Machine learning theory**;

KEYWORDS

robust statistics, list-decodable learning, learning mixtures of spherical Gaussians

ACM Reference Format:
Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. 2018. List-Decodable Robust Mean Estimation and Learning Mixtures of Spherical Gaussians. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3188745.3188758

1 INTRODUCTION

1.1 Background

This paper is concerned with the problem of efficiently learning high-dimensional spherical Gaussians in the presence of a large fraction of corrupted data, and in the related problem of parameter estimation for mixtures of high-dimensional spherical Gaussians (henceforth, spherical GMMs). Before we state our main results, we describe and motivate these two fundamental learning problems.

The first problem we study is the following:

Problem 1: List-Decodable Gaussian Mean Estimation. Given a set T of points in $\mathbb{R}^n$ and a parameter $\alpha \in (0, 1/2]$ with the promise that an α-fraction of the points in T are drawn from $G \sim N(\mu, I)$ — an unknown mean, identity covariance Gaussian — we want to output a "small" list of candidate vectors $\{\widehat{\mu}_1, \ldots, \widehat{\mu}_s\}$ such that at least one of the $\widehat{\mu}_i$'s is "close" to the mean μ of G, in Euclidean distance.

A few remarks are in order: We first note that we make no assumptions on the remaining $(1-\alpha)$-fraction of the points in T. These points can be arbitrary and may be chosen by an adversary that is computationally unbounded and is allowed to inspect the

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188758

set of good points. We will henceforth call such a set of points α-*corrupted*. Ideally, we would like to output a single hypothesis vector $\widehat{\mu}$ that is close to μ (with high probability). Unfortunately, this goal is information-theoretically impossible when the fraction α of good samples is less than $1/2$. For example, if the input distribution is a uniform mixture of $1/\alpha$ many Gaussians whose means are pairwise far from each other, there are $\Theta(1/\alpha)$ different valid answers and the list must by definition contain approximations to each of them. It turns out that the information-theoretically best possible size of the candidates list is $s = \Theta(1/\alpha)$. Therefore, the feasible goal is to design an efficient algorithm that minimizes the Euclidean distance between the unknown μ and its closest $\widehat{\mu}_i$.

The second problem we consider is the familiar task of learning the parameters of a spherical GMM. Let us denote by $N(\mu, \Sigma)$ the Gaussian with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n\times n}$. A Gaussian is called *spherical* if its covariance is a multiple of the identity, i.e., $\Sigma = \sigma^2 \cdot I$, for $\sigma \in \mathbb{R}_+$. An n-dimensional *k-mixture of spherical Gaussians* (spherical k-GMM) is a distribution on $\mathbb{R}^n$ with density function $F(x) = \sum_{i=1}^k w_i N(\mu_i, \sigma_i^2 \cdot I)$, where $w_i, \sigma_i \geq 0$, and $\sum_{i=1}^k w_i = 1$.

Problem 2: Parameter Estimation for Spherical GMMs. Given $k \in \mathbb{Z}_+$, a specified accuracy $\delta > 0$, and samples from a spherical k-GMM $F(x) = \sum_{i=1}^k w_i N(\mu_i, \sigma_i^2 \cdot I)$ on $\mathbb{R}^n$, we want to estimate the parameters $\{(w_i, \mu_i, \sigma_i), i \in [k]\}$ up to accuracy δ. More specifically, we want to return a list $\{(u_i, v_i, s_i), i \in [k]\}$ so that for some permutation $\pi \in \mathrm{S}_k$, we have that for all $i \in [k]$: $|w_i - u_{\pi(i)}| \leq \delta$, $\|\mu_i - v_{\pi(i)}\|_2/\sigma_i \leq \delta/w_i$, and $|\sigma_i - s_{\pi(i)}|/\sigma_i \leq (\delta/w_i)/\sqrt{n}$.

If $F' = \sum_{i=1}^k u_i N(v_i, s_i^2 \cdot I)$ is the hypothesis distribution, the above definition implies that $d_{\mathrm{TV}}(F, F') = O(k\delta)$. We will also be interested in the *robust* version of Problem 2. This corresponds to the setting when the input is an η-corrupted set of samples from a k-mixture of spherical Gaussians, where $\eta \ll \min_i w_i$.

Before we proceed with a detailed background and motivation, we point out the connection between these two problems. Intuitively, Problem 2 can be reduced to Problem 1, as follows: We can think of the samples drawn from a spherical GMM as a set of corrupted samples from a single Gaussian — where the Gaussian in question can be *any* of the mixture components. The output of the list decoding algorithm will produce a list of hypotheses with the guarantee that *every* mean vector in the mixture is relatively close to some hypothesis. If in addition the distances between the means and their closest hypotheses are substantially smaller than the distances between the means of different components, this will allow us to reliably cluster our sample points based on which hypothesis they are closest to. We can thus cluster points based on which component they came from, and then we can learn each component independently.

1.2 List-Decodable Robust Learning

The vast majority of efficient high-dimensional learning algorithms with provable guarantees make strong assumptions about the input data. In the context of unsupervised learning (which is the focus of this paper), the standard assumption is that the input points are independent samples drawn from a known family of generative models (e.g., a mixture of Gaussians). However, this simplifying assumption is rarely true in practice and it is important to design estimators that are *robust* to deviations from their model assumptions.

The field of robust statistics [27, 31] traditionally studies the setting where we can make *no* assumptions about a *"small" constant* fraction η of the data. The term "small" here means that $\eta < 1/2$, hence the input data forms a reasonably accurate representation of the true model. From the information-theoretic standpoint, robust estimation in this "small error regime" is fairly well understood. For example, in the presence of η-fraction of corrupted data, where $\eta < 1/2$, the Tukey median [48] is a robust estimator of location that approximates the mean of a high-dimensional Gaussian within ℓ_2-error $O(\eta)$ — a bound which is known to be information-theoretically best possible for *any* estimator. The catch is that computing the Tukey median can take exponential time (in the dimension). This curse of dimensionality in the running time holds for essentially all known estimators in robust statistics [8].

This phenomenon had raised the following question: *Can we reconcile computational efficiency and robustness in high dimensions?* Recent work in the TCS community made the first algorithmic progress on this front: Two contemporaneous works [13, 37] gave the first *computationally efficient* robust algorithms for learning high-dimensional Gaussians (and many other high-dimensional models) with error close to the information-theoretic optimum. Specifically, for the problem of robustly learning an unknown mean Gaussian $N(\mu, I)$ from an η-corrupted set of samples, $\eta < 1/2$, we now know a polynomial-time algorithm that achieves the information-theoretically optimal error of $O(\eta)$ [15].

The aforementioned literature studies the setting where the fraction of corrupted data is relatively small (smaller than $1/2$), therefore the real data is the *majority* of the input points. A related setting of interest focuses on the regime when the fraction α of real data is small — strictly smaller than $1/2$. From a practical standpoint, this "large error regime" is well-motivated by a number of pressing machine learning applications (see, e.g., [11, 44, 45]). From a theoretical standpoint, understanding this regime is of fundamental interest and merits investigation in its own right. A specific motivation comes from a previously observed connection to learning mixture models: Suppose we are given samples from the mixture $\alpha \cdot N(\mu, I) + (1-\alpha)E$, i.e., α-fraction of the samples are drawn from an unknown Gaussian, while the rest of the data comes from several other populations for which we have limited (or no) information. Can we approximate this "good" Gaussian component, independent of the structure of the remaining components?

More broadly, we would like to understand what type of learning guarantees are possible when the fraction α of good data is strictly less than $1/2$. While outputting a *single* accurate hypothesis is information-theoretically impossible, one may be able to efficiently compute a *small* list of candidate hypotheses with the guarantee that *at least one of them* is accurate. This is the notion of *list-decodable learning*, a model introduced by [6]. Very recently, [11] first studied the problem of robust high-dimensional estimation in the list-decodable model. In the context of robust mean estimation, [11] gave an efficient list-decodable learning algorithm with

the following performance guarantee: Assuming the true distribution of the data has bounded covariance, their algorithm outputs a list of $O(1/\alpha)$ candidate vectors one of which is guaranteed to achieve ℓ_2-error $\tilde{O}(\alpha^{-1/2})$ from the true mean.

Perhaps surprisingly, several aspects of list-decodable robust mean estimation are poorly understood. For example, is the $\tilde{O}(\alpha^{-1/2})$ error bound of the [11] algorithm best possible? If so, can we obtain significantly better error guarantees assuming additional structure about the real data? Notably — and in contrast to the small error regime — even basic *information-theoretic* aspects of the problem are open. That is, ignoring statistical and computational efficiency considerations, what is the minimum error achievable with $O(1/\alpha)$ (or $\mathrm{poly}(1/\alpha)$) candidate hypotheses for a given family of distributions?

The main focus of this work is on the fundamental setting where the good data comes from a Gaussian distribution. Specifically, we ask the following question:

Question 1.1. *What is the best possible error guarantee (information-theoretically) achievable for list-decodable mean estimation, when the true distribution is an unknown $N(\mu, I)$? More importantly, what is the best error guarantee that we can achieve with a computationally efficient algorithm?*

As our first main result, we essentially resolve Question 1.1.

1.3 Learning Mixtures of Separated Spherical Gaussians

A mixture of Gaussians or *Gaussian mixture model (GMM)* is a convex combination of Gaussian distributions, i.e., a distribution in $\mathbb{R}^n$ of the form $F = \sum_{i=1}^{k} w_i N(\mu_i, \Sigma_i)$, where the weights w_i, mean vectors μ_i, and covariance matrices Σ_i are unknown. GMMs are one of the most ubiquitous and extensively studied latent variable models in the literature, starting with the pioneering work of Karl Pearson [40]. In particular, the problem of parameter learning of a GMM from samples has received tremendous attention in statistics and computer science. (See Section 1.5 for a summary of prior work.)

In this paper, we focus on the natural and important case where each of the components is *spherical*, i.e., each covariance matrix is an unknown multiple of the identity. The majority of prior algorithmic work on this problem studied the setting where there is a minimum *separation* between the means of the components[1]. For the simplicity of this discussion, let us consider the case that the mixing weights are uniform (i.e., equal to $1/k$, where k is the number of components) and each component has identity covariance. (We emphasize that the positive results of this paper hold for the general case of an arbitrary mixture of high-dimensional spherical Gaussians, and apply even in the presence of a small dimension-independent fraction of corrupted data.) The problem of learning separated spherical GMMs was first studied by Dasgupta [12], followed by a long line of works that obtained efficient algorithms under weaker separation assumptions.

The currently best known algorithmic result in this context is the learning algorithm by Vempala and Wang [49] from 2002. Vempala and Wang gave a spectral algorithm with the following performance guarantee [49]: their algorithm uses $\mathrm{poly}(n, k, 1/\delta)$ samples and time, and learns a spherical k-GMM in n dimensions within parameter distance δ, as long as the pairwise distance (separation) between the component mean vectors is at least $k^{1/4}\mathrm{polylog}(nk/\delta)$. Obtaining a $\mathrm{poly}(n, k, 1/\delta)$ time algorithm for this problem that succeeds under weaker separation conditions has been an important open problem since.

Interestingly enough, until very recently, even the information-theoretic aspect of this problem was not understood. Specifically, what is the minimum separation that allows the problem to be solvable with $\mathrm{poly}(n, k, 1/\delta)$ samples? Recent work by Regev and Vijayraghavan [41] characterized this aspect of the problem: Specifically, [41] showed that the problem of learning spherical k-GMMs (with equal weights and identity covariances) can be solved with $\mathrm{poly}(n, k, 1/\delta)$ *samples* if and only if the means are pairwise separated by at least $\Theta(\sqrt{\log k})$. Unfortunately, the approach of [41] is non-constructive in high dimensions. Specifically, they gave a sample-efficient learning algorithm whose running time is exponential in the dimension. This motivates the following question:

Question 1.2. *Is there a $\mathrm{poly}(n, k)$ time algorithm for learning spherical k-GMMs with separation $o(k^{1/4})$, or better $O(k^{\varepsilon})$, for any fixed $\varepsilon > 0$? More ambitiously, is there an efficient algorithm that succeeds under the information-theoretically optimal separation?*

As our second main result, we make substantial progress towards the resolution of Question 1.2.

1.4 Our Contributions

In this paper, we develop a set of techniques that yield new efficient algorithms with significantly better guarantees for Problems 1 and 2. Our algorithms depend in an essential way on the analysis of high degree multivariate polynomials. We obtain a detailed structural understanding of the behavior of high degree polynomials under the standard multivariate Gaussian distribution, and leverage this understanding to design our learning algorithms. More concretely, our main technical contribution is a new technique, using degree-d multivariate polynomials, to remove outliers from high-dimensional datasets where the majority of the points are corrupted.

List-Decodable Mean Estimation. Our main result is an efficient algorithm for list-decodable Gaussian mean estimation with a significantly improved error guarantee:

Theorem 1.3 (List-Decodable Gaussian Mean Estimation). *Fix $d \in \mathbb{Z}_+$ and $0 < \alpha < 1$. There is an algorithm with the following performance guarantee: Given d, α, and a set $T \subset \mathbb{R}^n$ of cardinality $|T| = O(d^{2d}) \cdot n^{O(d)} / \mathrm{poly}(\alpha)$ with the promise that α-fraction of the points in T are independent samples from an unknown $G \sim N(\mu, I)$, $\mu \in \mathbb{R}^n$, the algorithm runs in time $O(nd/\alpha)^{O(d)}$ and with high probability outputs a list of $O(1/\alpha)$ vectors one of which is within ℓ_2-distance $\tilde{O}_d(\alpha^{-1/(2d)})$ of the mean μ of G.*

We note that the $\tilde{O}(\cdot)$ notation hides polylogarithmic factors in its argument. See Theorem 3.1 for a more detailed formal statement.

Discussion and Comparison to Prior Work. As already mentioned in Section 1.2, the only previously known algorithm for list-decodable

[1] Without any separation assumptions, it is known that the sample complexity of the problem becomes exponential in the number of components [28, 39].

mean estimation (for $\alpha < 1/2$) is due to [11] and achieves error $\tilde{O}(\alpha^{-1/2})$ under a bounded covariance assumption for the good data. As we will show later in this section (Theorem 1.5), this error bound is information-theoretically (essentially) best possible under such a second moment condition. Hence, additional assumptions about the good data are necessary to obtain a stronger bound. It should also be noted that the algorithm [11] does not lead to a better error bound, even for the case that the good distribution is an identity covariance Gaussian[2].

Our algorithm establishing Theorem 1.3 achieves substantially better error guarantees under stronger assumptions about the good data. The parameter d quantifies the tradeoff between the error guarantee and the sample/computational complexity of our algorithm. Even though it is not stated explicitly in Theorem 1.3, we note that for $d = 1$ our algorithm straightforwardly extends to all subgaussian distributions (with parameter $\nu = O(1)$), and gives error $\tilde{O}(\alpha^{-1/2})$. We also remark that our algorithm is spectral — in contrast to [11] that relies on semidefinite programming — and it may be practical for small constant values of d.

There are two important parameter regimes we would like to highlight: First, for $d = O(1/\varepsilon)$, where $\varepsilon > 0$ is an arbitrarily small constant, Theorem 1.3 yields a polynomial time algorithm that achieves error of $O(\alpha^{\varepsilon})$. Second, for $d = \Theta(\log(1/\alpha))$, Theorem 1.3 yields an algorithm that runs in time $(n/\alpha)^{O(\log(1/\alpha))}$ and achieves error of $\tilde{O}(\log^{3/2}(1/\alpha))$. This error bound comes close to the information-theoretic optimum of $\Theta(\sqrt{\log(1/\alpha)})$, established in Theorem 1.5. In the full version of this paper, we show that an adaptation of our algorithm works under the optimal separation of $O(\sqrt{\log(1/\alpha)})$.

A natural question is whether there exists a $\mathrm{poly}(n/\alpha)$ time list-decodable mean estimation algorithm with error $\mathrm{polylog}(1/\alpha)$, or even $\Theta(\sqrt{\log(1/\alpha)})$. In Theorem 1.6, we prove a Statistical Query (SQ) lower bound suggesting that the existence of such an algorithm is unlikely. More specifically, our SQ lower bound gives evidence that the complexity of our algorithm is qualitatively best possible.

High-Level Overview of Technical Contributions. Let $G \sim N(\mu, I)$ be the unknown mean Gaussian from which the α-fraction of good samples S are drawn, and T be the α-corrupted set of points given as input. We design an algorithm that iteratively detects and removes outliers from T, until we are left with a collection of $s = O(1/\alpha)$ many subsets $T_1, \ldots, T_s$ of T one of which is substantially "cleaner" than T. Specifically, the empirical mean of at least one of the T_i's will be $\tilde{O}_d(\alpha^{-1/(2d)})$ close to the unknown mean μ of G. Our algorithm is "spectral" in the sense that it works by analyzing the eigendecomposition of certain matrices constructed from degree-d moments of the empirical distribution. Specifically, to achieve error of $\tilde{O}_d(\alpha^{-1/(2d)})$, the algorithm of Theorem 1.3 works with matrices of dimension $O(n^d) \times O(n^d)$.

At a very high-level, our approach bears a similarity to the "filter" method — a spectral technique to iteratively detect and remove outliers from a dataset — introduced in [13], for efficient robust estimation in the "small error regime" (corresponding to $\alpha \gg 1/2$). Specifically, our algorithm tries to identify degree-d polynomials $p : \mathbb{R}^n \to \mathbb{R}$ such that the behavior of p on the corrupted set of samples T is significantly different from the expected behavior of p on the good set of samples S. One way to achieve this goal [13, 17] is by finding polynomials p with unexpectedly large empirical variance. The hope is that if we find such a polynomial, we can then use it to identify a set of points with a large fraction of corrupted samples and remove it to clean up our data set. This idea was previously used for robust estimation in the small error regime.

A major complication that occurs in the regime of $\alpha < 1/2$ is that since fewer than half of our samples are good, the values of such a polynomial p might concentrate in several clusters. As a consequence, we will not necessarily be able to identify which cluster contains the good samples. In order to deal with this issue, we need to develop new techniques for outlier removal that handle the setting that the good data is a small fraction of our dataset. Roughly speaking, we achieve this by performing a suitable clustering of points based on the values of p, and returning multiple (potentially overlapping) subsets of our original dataset T with the guarantee that at least one of them will be a cleaner version of T. This new paradigm for performing outlier removal in the large error regime may prove useful in other contexts as well.

A crucial technical contribution of our approach is the use of degree more than one polynomials for outlier removal in this setting. The intuitive reason for using polynomials of higher degree is this: A small fraction of points that are far from the true mean in some particular direction will have a more pronounced effect on higher degree moments. Therefore, taking advantage of the information contained in higher moments should allow us to discern smaller errors in the distance from the true mean. The difficulty is that it is not clear how to algorithmically exploit the structure of higher degree moments in this setting.

The major obstacle is the following: Since we do not know the mean μ of G — this is exactly the quantity we are trying to approximate! — we are also not able to evaluate the variance $\mathrm{Var}[p(G)]$ of $p(G)$. If p was a degree-1 polynomial, this would not be a problem, as the variance $\mathrm{Var}[p(G)]$ does not depend on μ. But for degree at least 2 polynomials, the dependence of $\mathrm{Var}[p(G)]$ on μ becomes a fundamental difficulty. Thus, although we can potentially find polynomials with unexpectedly large empirical variance, we will have no way of knowing whether this is due to corrupted points $x \in T$ (on which $p(x)$ is abnormally far from its true mean), or due to errors in our estimation of the mean of G causing us to underestimate the variance $\mathrm{Var}[p(G)]$.

In order to circumvent this difficulty, we require a number of new ideas, culminating in an algorithm that allows us to either verify that the variance of $p(G)$ is close to what we are expecting, or to find some other polynomial that allows us to remove outliers.

Learning Mixtures of Separated Spherical GMMs. We leverage the connection between list-decodable learning and learning mixture models to obtain an efficient algorithm for learning spherical GMMs under much weaker separation assumptions. Specifically, by using the algorithm of Theorem 1.3 combined with additional algorithmic ideas, we obtain our second main result:

Theorem 1.4 (Learning Separated Spherical GMMs). *There is an algorithm with the following performance guarantee: Given $d \in \mathbb{Z}_+$, $\alpha > 3\delta \geq 0$, and sample access to a k-mixture of spherical*

[2] Intuitively, this holds because the [11] algorithm only uses the first two empirical moments. It can be shown that more moments are necessary to improve on the $O(\alpha^{-1/2})$ error bound (see the construction in the proof of Theorem 1.6).

Gaussians $F = \sum_{i=1}^{k} w_i N(\mu_i, \sigma_i^2 I)$ *on* $\mathbb{R}^n$, *where* $n = \Omega(\log(1/\alpha))$, *with* $w_i \geq \alpha$ *for all* i, *and so that* $\|\mu_i - \mu_j\|_2/(\sigma_i + \sigma_j)$ *is at least*

$$S \stackrel{\text{def}}{=} C\left(\alpha^{-1/(2d)}\sqrt{d}(d + \log(1/\alpha))\log(2 + \log(1/\alpha))^2 + \sqrt{\log(k/\delta)}\right),$$

for all $i \neq j$, *for* $C > 0$ *a sufficiently large constant, the algorithm draws* $\mathrm{poly}(n, (dk/\delta)^d)$ *samples from* F, *runs in time* $\mathrm{poly}(n, (dk/\delta)^d)$, *and with high probability returns a list* $\{(u_i, v_i, s_i), i \in [k]\}$, *such that the following conditions hold (up to a permutation):* $|u_i - w_i| = O(\delta)$, $\|\mu_i - v_i\|_2/\sigma_i = O(\delta/w_i)$, *and* $|s_i - \sigma_i|/\sigma_i = O(\delta/w_i)/\sqrt{n}$.

The reader is also referred to Proposition 4.3 for a more detailed statement that also allows a small, dimension-independent fraction of adversarial noise in the input samples.

Discussion and High-Level Overview. To provide a cleaner interpretation of Theorem 1.4, we focus on the prototypical case of a uniform mixture of identity covariance Gaussians. For this case, Theorem 1.4 reduces to the following statement (see Corollary 4.10): For any $\varepsilon > 0$, if the pairwise separation between the means is at least $\Omega(k^\varepsilon + \sqrt{\log(k/\delta)})$, our algorithm learns the parameters up to accuracy δ in time $\mathrm{poly}\left(n, 1/\delta, (k/\varepsilon)^{1/\varepsilon}\right)$. Prior to our work, the best known efficient algorithm [49] required separation $\Omega(k^{1/4} + \sqrt{\log(k/\delta)})$. Also note that by setting $d = \Theta(\log k)$, we obtain a learning algorithm with sample complexity and running time $\mathrm{poly}(n, 1/\delta, k^{\log k})$ that works with separation of $\tilde{O}(\log^{3/2}(k) + \sqrt{\log(1/\delta)})$. This separation bound comes close to the information-theoretically minimum of $\Omega(\sqrt{\log k})$ [41]. (We also note that improving the error bound in Theorem 1.3 to $O(\sqrt{\log(1/\alpha)})$, for $d = O(\log(1/\alpha))$, directly improves our separation bound to $O(\sqrt{\log k})$.)

We now provide an intuitive explanation of our spherical GMM learning algorithm. First, we note that we can reduce the dimension of the problem from n down to some function of k. When the covariance matrices of the components are nearly identical, this can be done with a twist of standard techniques. For the case of arbitrary covariances, we need to employ a few additional ingredients.

When each component has the same covariance matrix, the learning algorithm is quite simple: We start by running our list-decoding algorithm (Theorem 1.3) with appropriate parameters to get a small list of hypothesis means. We then associate each sample with the closest element of our list. At this point, we can cluster the points based on which means they are associated to and use this clustering to accurately learn the correct components.

The general case, when the covariances of the components are arbitrary, is significantly more complex. In this case, we can recover a list H of candidate means only after first guessing the radius of the component that we are looking for. Without too much difficulty, we can find a large list of guesses and thereby produce a list of hypotheses of size $\mathrm{poly}(n/\alpha)$. However, clustering based on this list now becomes somewhat more difficult, as we do not know the radius at which to cluster. We address this issue by performing a secondary test to determine whether or not the cluster that we have found contains many points at approximately the correct distance from each other.

Minimax Error Bounds and SQ Lower Bounds. As mentioned in Section 1.2, even the following information-theoretic aspect of list-decodable mean estimation is open: Ignoring sample complexity and running time, how small a distance from the true mean can be achieved with $\mathrm{poly}(1/\alpha)$ many hypotheses or number of hypotheses that is only a function of α, i.e., independent of the dimension n?

Theorem 1.3 implies that we can achieve error $\mathrm{polylog}(1/\alpha)$ for Gaussians. We show that the optimal error bound (upper and lower bound) for the case $N(\mu, I)$ and more generally for subgaussian distributions is in fact $\Theta(\sqrt{\log(1/\alpha)})$. Moreover, under bounded k-th moment assumptions, for even k, the optimal error is $\Theta_k(\alpha^{-1/k})$.

Theorem 1.5 (Minimax Error Bounds). *Let* $0 < \alpha < 1/2$. *There exists an (inefficient) algorithm that given a set of* α*-corrupted samples from a distribution* D, *where (a)* D *is subgaussian with bounded variance in each direction, or (b)* D *has bounded first* k *moments, for even* k, *outputs a list of* $O(1/\alpha)$ *vectors one of which is within distance* $g(\alpha)$ *from the mean* μ *of* D, *and* $g(\alpha) = O(\sqrt{\log(1/\alpha)})$ *in case (a) and* $g(\alpha) = O_k(\alpha^{-1/k})$ *in case (b). Moreover, these error bounds are optimal, up to constant factors. Specifically, the error bound of (a) cannot be asymptotically improved even if* $D = N(\mu, I)$, *as long as the list size is* $\mathrm{poly}(1/\alpha)$. *The error bound of (b) cannot be asymptotically improved as long as the list size is only a function of* α.

For the detailed statements, the reader is referred to the full version.

We now turn to our computational lower bounds. Given Theorem 1.5, the following natural question arises: For the case of Gaussians, can we achieve the minimax bound in polynomial time? We provide evidence that this may not be possible, by proving a Statistical Query (SQ) lower bound for this problem. Recall that a Statistical Query (SQ) algorithm [35] relies on an oracle that given any bounded function on a single domain element provides an estimate of the expectation of the function on a random sample from the input distribution. This is a restricted but broad class of algorithms, encompassing many algorithmic techniques in machine learning. A recent line of work [21–24] developed a framework of proving unconditional lower bounds on the complexity of SQ algorithms for search problems over distributions.

By leveraging this framework, using the techniques of our previous work [17], we show that any SQ algorithm for list-decodable Gaussian mean estimation that guarantees error $\alpha^{-1/d}$, for some $d \geq 2$, requires either high accuracy queries or exponentially many queries:

Theorem 1.6 (SQ Lower Bounds). *Any SQ list-decodable mean estimation algorithm for* $G \sim N(\mu, I)$ *that returns a list of sub-exponential size so that some element in the list is within distance* $O(\alpha^{-1/d})$ *of the mean* μ *of* G *requires either queries of accuracy* $2^{O((1/\alpha)^{2/d})} \cdot n^{-\Omega(d)}$ *or* $2^{n^{\Omega(1)}}$ *queries.*

The reader is referred to the full version for the formal statement and proof.

1.5 Related Work

Robust Estimation. The field of robust statistics [27, 31, 32, 42, 47] studies the design of estimators that are stable to model misspecification. After several decades of investigation, the statistics community has discovered a number of estimators that are provably

robust in the sense that they can tolerate a constant (less than 1/2) fraction of corruptions, independent of the dimension. While the information-theoretic aspects of robust estimation have been understood, the central algorithmic question — that of designing robust and computationally efficient estimators in high-dimensions — had remained open.

Recent work in computer science [13, 37] shed light to this question by providing the first efficient robust learning algorithms for a variety of high-dimensional distributions. Specifically, [13] gave the first robust learning algorithms that can tolerate a constant fraction of corruptions, independent of the dimension. Subsequently, there has been a flurry of research activity on algorithmic robust high-dimensional estimation. This includes robust estimation of graphical models [16], handling a large fraction of corruptions in the list-decodable model [11, 43], developing robust algorithms under sparsity assumptions [5], obtaining optimal error guarantees [15], establishing computational lower bounds for robust estimation [17], establishing connections with robust supervised learning [18], and designing practical algorithms for data analysis applications [14].

Learning GMMs. A long line of work initiated by Dasgupta [12], see, e.g., [2, 4, 10, 34, 49], provides computationally efficient algorithms for recovering the parameters of a GMM under various separation assumptions between the mixture components. More recently, efficient parameter learning algorithms were obtained [7, 28, 39] under minimal information-theoretic separation assumptions. Without separation conditions, the sample complexity of parameter estimation is known to scale exponentially with the number of components, even in one dimension [28, 39]. To circumvent this information-theoretic bottleneck of parameter learning, a related line of work has studied parameter learning in a smoothed setting [3, 9, 25, 26, 30]. The related problems of density estimation and proper learning for GMMs have also been extensively studied [1, 20, 28, 38, 39, 46]. In density estimation (resp. proper learning), the goal is to output some hypothesis (resp. GMM) that is close to the unknown mixture in total variation distance.

Most relevant to the current work is the classical work of Vempala and Wang [49] and the very recent work by Regev and Vijayraghavan [41]. Specifically, [49] gave an efficient algorithm that learns the parameters of *spherical* GMMs under the weakest separation conditions known to date. On the other hand, [41] characterize the separation conditions under which parameter learning for spherical GMMs can be solved with $\mathrm{poly}(n, k, 1/\delta)$ *samples*. Whether such a separation can be achieved with an efficient algorithm was left open in [41]. Our work makes substantial progress in this direction.

1.6 Concurrent Works

Two concurrent and independent works [29, 36] used the sum-of-squares hierarchy to obtain qualitatively similar algorithmic results to ours for learning mixtures of spherical Gaussians.

1.7 Detailed Overview of Techniques

1.7.1 List-Decodable Mean Estimation:

Outlier Removal and Challenges of the Large Error Regime. We start by reviewing the framework of [13] for robust mean estimation in the small error regime, followed by an explanation of the main difficulties that arise in the large error regime of the current paper.

In the small error regime, the "filtering" algorithm of [13] for robust Gaussian mean estimation works by iteratively detecting and removing outliers (corrupted samples) until the empirical variance in every direction is not much larger than expected. If every direction has small empirical variance, then the true mean and the empirical mean are close to each other [13]. Otherwise, the [13] algorithm projects the input points in a direction of maximum variance and throws away those points whose projections lie unexpectedly far from the empirical median in this direction. While this iterative spectral technique for outlier removal is by now well-understood for the small error regime (and has been applied to various settings), there are two major obstacles that arise if one wants to generalize it to the large error regime, i.e., where only a small fraction α of samples are good.

The first difficulty is that even the one-dimensional version of the problem in the large error regime is non-trivial. Specifically, consider a direction v of large empirical variance. The [13] algorithm exploits the fact that the empirical median is a robust estimator of the mean in the one-dimensional setting. In contrast, in the large error regime, it is not clear how to approximate the true mean of a one-dimensional projection. This holds for the following reason: The input distribution can simulate a mixture of $1/\alpha$ many Gaussians whose means are far from each other, and the algorithm will have no way of knowing which is the real one. In order to get around this obstacle, we construct more elaborate outlier-removal algorithms, which we call *multifilters*. Roughly speaking, a multifilter can return several (potentially overlapping) subsets of the original dataset T with the guarantee that *at least one* of these subsets is substantially "cleaner" than T.

The second difficulty is somewhat harder to deal with. As already mentioned, the filtering algorithm of [13] iteratively removes outliers by looking for directions in which the empirical distribution has a substantially larger variance than it should. In the low error regime, this approach does a good job of detecting are removing the corrupted points that can move the empirical mean far from the true mean. In the large error regime, the situation is substantially different. In particular, it is entirely possible that the empirical distribution does not have abnormally large variance in *any* direction, while still the empirical mean is $\Omega(\sqrt{1/\alpha})$-far from the true mean. That is, considering the variance of one-dimensional projections of our dataset in various directions seems inadequate in order to improve the $O(\sqrt{1/\alpha})$ error bound. This obstacle is inherent: the variance of *linear polynomials* (projections) is not a sufficiently accurate method of detecting a small fraction of good samples being substantially displaced from the mean of the bad samples. To circumvent this obstacle, we will use *higher degree polynomials*, which are much more sensitive to a small fraction of points being far away from the others. In particular, our algorithms will search for degree-d polynomials that have abnormally large expectation or variance, and use such polynomials to construct our multifilters.

Overview of List-Decodable Mean Estimation Algorithm. The basic overview of our algorithm is as follows: We compute the sample mean μ_T of the α-corrupted set T, and then search for (appropriate) degree-d polynomials whose empirical expectation or variance is

too large relative to what it should be, assuming that the good distribution G is $N(\mu_T, I)$ — an identity covariance Gaussian with mean μ_T. We note that this task can be done efficiently with an eigenvalue computation, by taking advantage of the appropriate orthogonal polynomials. If there are no degree-d polynomials with too large variance, we can show that the sample mean μ_T is within distance $\tilde{O}_d(\alpha^{-1/(2d)})$ from the true mean. On the other hand, if we do find a degree-d polynomial with abnormally large variance, we will be able to produce a multifilter and make progress.

We now sketch how to exploit the existence of a large variance polynomial p to construct a multifilter. Intuitively, the existence of such a polynomial p suggests that there are many points that are far away from other points, and therefore separating these points into (potentially overlapping) clusters should guarantee that almost all good points are in the same cluster. Unfortunately, for this idea to work, we need to know that the variance of p on the good set of points S is not too large. For degree-1 polynomials p this condition holds automatically. If S is a sufficiently large set of samples from $G \sim N(\mu, I)$ and p is a normalized linear form, then $\mathrm{Var}[p(S)] \approx \mathrm{Var}[p(G)] = 1$. But if p has degree at least 2, the variance $\mathrm{Var}[p(S)]$ depends on the true mean μ, which unfortunately is unknown. Fortunately, there is a way to circumvent this obstacle by either producing a multifilter or verifying that the variance $\mathrm{Var}[p(G)]$ is not too large.

We do this as follows: Firstly, we show that the variance $\mathrm{Var}[p(G)]$, $G \sim N(\mu, I)$, can be expressed as an average of $p_i^2(\mu)$ for some explicitly computable, normalized, homogeneous polynomials p_i. We then need to algorithmically verify that the polynomials $p_i(\mu)$ are not too large. This is difficult to do directly, so instead we replace each p_i by the corresponding *multilinear* polynomial q_i, and note that $p_i(\mu)$ is the average value of q_i at many independent copies of G. If this is large, then it means that evaluating q_i at a random tuple of samples will often have larger than expected size.

This idea will allow us to produce a multifilter for the following reason: Since each q_i is multilinear, this essentially allows us to write it as a composition of linear functions. More rigorously, we use the following iterative process: We iteratively plug-in variables one at a time to q_i. If at any step the size of the resulting polynomial jumps substantially, then the fact that this size is not well-concentrated as we try different samples will allow us to produce a multifilter.

1.7.2 Learning Spherical GMMs:

The Identity Covariance Case. Since a Gaussian mixture model can simultaneously be thought of as a mixture of any one of its components with some error distribution, applying our list-decoding algorithm to samples from a GMM will return a list of hypotheses so that *every* mean in the mixture is close to some hypothesis in the list. We can then use this list to cluster our samples by component.

In particular, given samples from a Gaussian $G = N(\mu, I)$ and many possible means $h_1, \ldots, h_m$, we consider the process of associating a sample x from G with the nearest h_i. We note that x is closer to h_j than h_i if and only if its projection onto the line between them is. Now if h_i is substantially closer to μ than h_j is, then this requires that this projection (which is Gaussian distributed) be far from its mean, which happens with tiny probability. Thus, by a union bound, as long as our list contains some h_i that is close to μ, the closest hypothesis to x with high probability is not much further. If the separation between the means in our mixture is much larger than the separation between the means and the closest hypotheses, this implies that almost all samples are associated with one of the hypotheses near its component mean, and this will allow us to cluster samples by component. This idea of clustering points based on which of a finite set they are close to is an important idea that shows up in several related contexts in this paper.

The General Case. The above idea works more or less as stated for mixtures of identity covariance Gaussians, but when dealing with more general mixtures of spherical Gaussians several complications arise. Firstly, in order to run out list-decoding algorithm, we need to know (a good approximation to) the covariance matrix of each component. The other difficulty is that, in order to cluster points, we will take a set of all nearby hypotheses that have reasonable numbers of samples associated with them. The issue is that we no longer know what "nearby" means, as it should depend on the covariance matrix of the associated Gaussian.

To solve the first of these problems we use a trick that will be reused several times. We note that two samples from the same Gaussian $N(\mu, \sigma^2 I)$ have distance approximately $\Theta(\sigma\sqrt{n})$, and that even one sample from $N(\mu, \sigma^2 I)$ is unlikely to be much closer than this to samples from different components. Therefore, by simply looking at the distance to the closest other sample gives us a constant factor approximation to the standard deviation of the corresponding component. This allows us to write down a polynomial-size list of viable hypothesis standard deviations. Running our list decoding algorithm for each standard deviation, gives us a polynomial-size list of hypothesis means.

To solve the second problem, we use the above idea to approximate the standard deviations associated to our sample points. When clustering them, we look for collections of sample points with standard deviations approximately the same σ, whose closest hypotheses are within some reasonable multiple of σ of each other. Since we are able to approximate the size of the component that our samples are coming from, we can guarantee that we aren't accidentally merging several smaller clusters together by using the wrong radius.

Dimension Reduction. One slight wrinkle with the above sketched learning algorithm is that since the number of candidate hypotheses is polynomial in n, the separation between the components will be required to be at least $\sqrt{\log(n)}$. This bound is suboptimal, when n is very large. Another issue is that the overall runtime of the learning algorithm would not be a fixed polynomial in n, but would scale as n^d. There is a way around both these issues, by reducing to a lower dimensional problem.

In particular, standard techniques involve looking at the k largest principle values that allow one to project onto a subspace of dimension k without losing too much. Unfortunately, these ideas require that all of the Gaussians involved have roughly the same covariance. Fortunately, if n is large, our ability to approximate the covariance associated to a sample by looking at its distances to other samples becomes more accurate. Using a slightly modification of this idea, we can actually break our samples into subsets so that each subset

is a mixture of Gaussians of approximately the save covariance. By projecting each of these in turn, we can reduce the original problem to a poly(k) number of dimensions and eliminate this extra term.

1.7.3 Minimax Error Bounds: We now explain our approach to pin down the information-theoretic optimal error for the list-decodable mean estimation problem. Concretely, for the identity covariance Gaussian case we show that there is an (inefficient) algorithm that guarantees that some hypothesis is within $O(\sqrt{\log(1/\alpha)})$ of the true mean. The basic idea is that the true mean must have the property that there is an α-fraction of samples that are well-concentrated (in the sense of having good tail bounds in every direction) about the point. The goal of our (inefficient) algorithm will be to find a small number of balls of radius $O(\sqrt{\log(1/\alpha)})$ that covers the set of all such points. We show that such a set exists using the covering/packing duality. In particular, we note that if there are a large number of such sets with means far apart, we get a contradiction since the sets must be individually large but their overlaps must be pairwise small (due to concentration bounds).

This approach immediately generalizes to provide a list-decodable mean estimation algorithm for any distribution with known tail bounds, providing an error $O(t)$, where only an α-fraction of the points are more than t-far from the mean in any direction. This generic statement has a number of implications for various families. In particular, it gives a (tight) error upper bound of $O(\sqrt{\log(1/\alpha)})$ for subgaussian distributions with bounded variance in each direction. Previously, no upper bound better than $\tilde{O}(1/\sqrt{\alpha})$ was known for these families. For distributions whose first k central moments are bounded from above (for even k), we obtain a tight error upper bound of $O_k(\alpha^{-1/k})$.

Regarding lower bounds, [11] showed an $\Omega(\sqrt{\log(1/\alpha)})$ error lower bound for $N(\mu, \Sigma)$, where Σ is *unknown* and $\Sigma \preceq I$. We strengthen this result by showing that the $\Omega(\sqrt{\log(1/\alpha)})$ lower bound holds even for $N(\mu, I)$. We also prove matching lower bounds of $\Omega_k(\alpha^{-1/k})$ for distributions with bounded moments. Our proofs proceed by exhibiting distributions X, so that X can be written as $X = \alpha X_i + (1-\alpha)E_i$ for many different X_i satisfying the necessary hypotheses. Then any list-decoding algorithm must return a list of hypotheses close to the mean of *every* X_i. If there are many such X_i's with means pairwise separated, then the list-decoding algorithm must either return many hypotheses or have large error.

1.7.4 SQ Lower Bounds: Finally, we prove lower bounds for list-decoding algorithms in the Statistical Query (SQ) model. Roughly speaking, we show that any SQ algorithm must either spend n^d time or have accuracy higher than $\alpha^{-1/2d}$, suggesting that our list-decoding algorithm is qualitatively tight in its tradeoff between runtime and sample complexity.

We prove these bounds using the technology developed in [17]. This basically reduces to finding a 1-dimensional distribution whose first many moments agree with the corresponding moments of a standard Gaussian. In our case, this amounts to constructing a one-dimensional distribution $A = \alpha N(\alpha^{-1/d}, 1) + (1-\alpha)E$, so that A's first d moments agree with those of a standard Gaussian. This can be done essentially because the $\alpha N(\alpha^{-1/d}, 1)$ part of the distribution only contributes at most a constant to any of the first d moments. This allows us to take E approximately Gaussian but slightly tweaked near 0 in order to fix these first few moments.

We note however, that if we move the error component much further from 0, its contribution to the d^{th} moment becomes super-constant and thus impossible to hide. This corresponds to the fact that degree-d moments are sufficient (and necessary) in order to detect errors of size $\alpha^{-1/d}$.

1.8 Organization

The structure of this extended abstract is as follows: In Section 2, we provide the necessary definitions and technical facts. In Section 3, we present a detailed overview of our list-decoding algorithm. Section 4 gives our algorithm for GMMs. Due to space limitations, most proofs are deferred to the full version [19].

2 DEFINITIONS AND PRELIMINARIES

2.1 Notation and Basic Definitions

Notation. For $n \in \mathbb{Z}_+$, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. If v is a vector, let $\|v\|_2$ denote its Euclidean norm. If M is a matrix, let $\|M\|_F$ denote its Frobenius norm.

Our algorithm and its analysis will make essential use of tensor analysis. For a tensor A, we will denote by $\|A\|_2$ the ℓ_2-norm of its entries.

Let $T \subset \mathbb{R}^n$ be a finite multiset. We will use $X \in_u T$ to denote that X is drawn uniformly from T. For a function $f : \mathbb{R}^n \to \mathbb{R}$, we will denote by $f(T)$ the random variable $f(X)$, $X \in_u T$.

Our basic objects of study are the Gaussian distribution and finite mixtures of spherical Gaussians:

Definition 2.1. *The n-dimensional* Gaussian $N(\mu, \Sigma)$ *with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n \times n}$ is the distribution with density function $f(x) = (2\pi)^{-n/2} \det(\Sigma)^{-1/2} \exp(-(1/2)(x-\mu)^T \Sigma^{-1}(x-\mu))$. A Gaussian is called* spherical *if its covariance is a multiple of the identity, i.e., $\Sigma = \sigma^2 \cdot I$, for $\sigma \in \mathbb{R}_+$.*

Definition 2.2. *An n-dimensional k-*mixture of spherical Gaussians *(spherical k-GMM) is a distribution on $\mathbb{R}^n$ with density function $F(x) = \sum_{j=1}^k w_j N(\mu_j, \sigma_j^2 \cdot I)$, where $w_j \geq 0$, $\sigma_j \geq 0$, for all j, and $\sum_{j=1}^k w_j = 1$.*

Definition 2.3. *The* total variation distance *between two distributions (with probability density functions) $P, Q : \mathbb{R}^n \to \mathbb{R}_+$ is defined to be $d_{\mathrm{TV}}(P, Q) \stackrel{\mathrm{def}}{=} (1/2) \cdot \|P - Q\|_1 = (1/2) \cdot \int_{x \in \mathbb{R}^n} |P(x) - Q(x)| dx$.*

2.2 Formal Problem Definitions

We record here the formal definitions of the problems that we study. Our first problem is robust mean estimation in the list-decodable learning model. We start by defining the list-decodable model:

Definition 2.4 (List Decodable Learning, [6]). *We say that a learning problem is (m, ε)-*list decodably solvable *if there exists an efficient algorithm that can output a set of m hypotheses with the guarantee that at least one is accurate to within error ε with high probability.*

Our notion of robust estimation relies on the following model of corruptions:

Definition 2.5 (Corrupted Set of Samples). *Given $0 < \alpha \leq 1$ and a distribution family $\mathcal{D}$, an α-*corrupted set of samples *T of size m is generated as follows: First, a set S of $\alpha \cdot m$ many samples are drawn independently from some unknown $D \in \mathcal{D}$. Then an omniscient adversary, that is allowed to inspect the set S, adds an arbitrary set of $(1-\alpha) \cdot m$ many points to the set S to obtain the set T.*

We are now ready to define the problem of list-decodable robust mean estimation:

Definition 2.6 (List-Decodable Robust Mean Estimation). *Fix a family of distributions $\mathcal{D}$ on $\mathbb{R}^n$. Given a parameter $0 < \alpha \leq 1$ and an α-corrupted set of samples T from an unknown distribution $D \in \mathcal{D}$, with unknown mean $\mu \in \mathbb{R}^n$, we want to output a list of $s = \mathrm{poly}(1/\alpha)$ candidate mean vectors $\widehat{\mu}_1, \ldots, \widehat{\mu}_s$ such that with high probability it holds $\min_{j=1}^{s} \|\widehat{\mu}_j - \mu\|_2 = g(\alpha)$, for some function $g : \mathbb{R} \to \mathbb{R}$. We say that $g(\alpha)$ is the* error guarantee *achieved by the algorithm.*

Our main algorithmic result is for the important special case that $\mathcal{D}$ is the family of unknown mean known covariance Gaussian distributions. We also establish minimax bounds that apply for more general distribution families.

Our second problem is that of learning mixtures of separated spherical Gaussians:

Definition 2.7 (Parameter Estimation for Spherical GMMs). *Given a positive integer k and samples from a spherical k-GMM $F(x) = \sum_{i=1}^{k} w_i N(\mu_i, \sigma_i^2 \cdot I)$, we want to estimate the parameters $\{(w_i, \mu_i, \sigma_i), i \in [k]\}$ up to a required accuracy δ. More specifically, we would like to return a list $\{(u_i, v_i, s_i), i \in [k]\}$ so that with high probability the following holds: For some permutation $\pi \in \mathrm{S}_k$ we have that for all $i \in [k]$: $|w_i - u_{\pi(i)}| \leq \delta$, $\|\mu_i - v_{\pi(i)}\|_2/\sigma_i \leq \delta/w_i$, and $|\sigma_i - s_{\pi(i)}|/\sigma_i \leq (\delta/w_i)/\sqrt{n}$.*

The above approximation of the parameters implies that

$$d_{\mathrm{TV}}(\sum_{i=1}^{k} w_i N(\mu_i, \sigma_i^2 I), \sum_{i=1}^{k} u_i N(v_i, s_i^2 I)) = O(k\delta) \ .$$

The sample complexity (hence, also the computational complexity) of parameter estimation depends on the smallest weight $\min_i w_i$ and the minimum separation between the components.

2.3 Basics of Hermite Analysis and Concentration

We briefly review the basics of Hermite analysis over $\mathbb{R}^n$ under the standard n-dimensional Gaussian distribution $N(0, I)$. Consider $L^2(\mathbb{R}^n, N(0, I))$, the vector space of all functions $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathbb{E}_{x \sim N(0,I)}[f(x)^2] < \infty$. This is an inner product space under the inner product

$$\langle f, g \rangle = \mathbb{E}_{X \sim N(0,I)}[f(X)g(X)] \ .$$

This inner product space has a complete orthogonal basis given by the *Hermite polynomials*. For univariate degree-i Hermite polynomials, $i \in \mathbb{N}$, we will use the *probabilist's* Hermite polynomials, denoted by $\mathrm{He}_i(x)$, $x \in \mathbb{R}$, which are scaled to be *monic*, i.e., the lead term of $\mathrm{He}_i(x)$ is x^i. For $a \in \mathbb{N}^n$, the n-variate Hermite polynomial $\mathrm{He}_a(x)$, $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, is of the form $\prod_{i=1}^{n} \mathrm{He}_{a_i}(x_i)$, and has degree $\|a\|_1 = \sum a_i$. These polynomials form a basis for the vector space of all polynomials which is orthogonal under this inner product. For a polynomial $p : \mathbb{R}^n \to \mathbb{R}$, its L^2-norm is $\|p\|_2 \stackrel{\mathrm{def}}{=} \sqrt{\langle p, p \rangle} = \mathbb{E}_{X \sim N(0,I)}[p(X)^2]^{1/2}$.

We will need the following standard concentration bound for degree-d polynomials over independent Gaussians (see, e.g., [33]):

Fact 2.8 ("degree-d Chernoff bound"). *Let $G \sim N(\mu, I)$, $\mu \in \mathbb{R}^n$. Let $p : \mathbb{R}^n \to \mathbb{R}$ be a real degree-d polynomial. For any $t > 0$, we have that $\Pr\left[|p(G) - \mathbb{E}[p(G)]| \geq t \cdot \sqrt{\mathrm{Var}[p(G)]}\right] \leq \exp(-\Omega(t^{2/d}))$.*

3 LIST-DECODABLE ROBUST MEAN ESTIMATION ALGORITHM

In this section, we provide a detailed outline of our main algorithmic result on list-decodable mean estimation:

Theorem 3.1 (List-Decodable Mean Estimation). *There exists an algorithm* `List-Decode-Gaussian` *that, given $0 < \alpha < 1/2$, $d \in \mathbb{Z}_+$, a failure probability $\tau > 0$, and a set T of $O(d!^2 \cdot n^{4d} \cdot \log(1/\tau)/\alpha^7)$ points in $\mathbb{R}^n$, of which at least a 2α-fraction are independent samples from a Gaussian $G \sim N(\mu, I)$, runs in time $(nd \log(1/\tau)/\alpha)^{O(d)}$ and returns a list of $O(1/\alpha)$ points such that, with probability at least $1 - \tau$, the list contains an $x \in \mathbb{R}^n$ with*

$$\|x - \mu\|_2 = O\left(\alpha^{-1/(2d)} \sqrt{d}(d + \log(1/\alpha)) \log(2 + \log(1/\alpha))^2\right) \ .$$

Detailed Structure of Algorithm. The key idea procedure behind our algorithm is a subroutine that given a set of samples either cleans it up producing one or two subsets at least one of which has substantially fewer errors than the original, or certifies that the mean of G must be close to the empirical mean. Using this subroutine, our final algorithm can be obtained by repeatedly applying the subroutine recursively to the returned sets until they produce vectors.

Before we can get into the detailed overview of this proof, it is necessary to lay out some technical groundwork. First, we will want to have a deterministic condition under which our algorithm will succeed. To that end, we introduce two important definitions. We say that a set S is representative of G if it behaves like a set of independent samples of G, in particular in the sense that it is a PRG against low-degree polynomial threshold functions for G. We also say that a larger set T is good if (roughly speaking) an α-fraction of the elements of T are a representative set for G. For technical reasons, will will also want the points of T to be not too far apart from each other.

We show that given a large set of points that contain an α-fraction of good points from, one can algorithmically find $O(1/\alpha)$ many subsets so that with high probability at least one of them is good (and thus can be fed into the rest of our algorithm). This would be immediate if it were it not for the requirement that the points in a good set be not too far apart. As it stands, this will require that we perform some very basic clustering algorithms.

The actual design of our multifilter involves working with several types of "pure" degree-d polynomials and their appropriate tensors. In particular, we need to pay attention to harmonic polynomials (which behave well with respect to L^2-norms), homogeneous polynomials, and multilinear polynomials.

The multifilter at its base level requires a routine that given a polynomial p, where $p(T)$ behaves very differently from $p(G)$,

allows us to use the values of $p(x)$ to separate the points coming from G from the errors. The basic idea of the technique is to cluster the points $p(x)$, for $x \in T$, and throw away points that are too far from any cluster large enough to correspond to the bulk of the values of $p(G)$ (which must be well-concentrated), or to divide T into two subsets with enough overlap to guarantee that any such cluster could be entirely contained on one side or the other.

Given this basic multifilter, the high-level picture for our main subroutine is as follows: Using spectral methods we can find if there are any degree-d polynomials p where $\mathbb{E}[p(T)^2]$ is substantially larger than it should be if T consisted of samples from $N(\mu_T, I)$. If there are no such polynomials, it is not hard to see that $\|\mu - \mu_T\|_2$ is small giving us our desired approximation. Otherwise, we would like to apply our basic multifilter algorithm to get a refined version of T.

Unfortunately, the application of the multifilter in the application above has a slight catch to it. Our basic multifilter will only apply to p if we can verify that $\mathrm{Var}[p(G)]$ is not too large. This would be easy to verify if we knew the mean of G, but unfortunately, we do not and errors in our approximation may lead to $\mathrm{Var}[p(G)]$ being much larger than anticipated, and in fact, potentially too large to apply our filter productively. In order to correct this, we will need new techniques to either prove that $\mathrm{Var}[p(G)]$ is small or to find a filter in the process. Using analytic techniques, we show that the $\mathrm{Var}[p(G)]$ is a weighted average of squares of $q_i(\mu - \mu_T)$ for some normalized, homogeneous polynomials q_i. Thus, it suffices to verify that each $q_i(\mu - \mu_T)$ is small.

To deal with this issue, it is actually much easier to work with multilinear polynomials, and so instead we deal with multilinear polynomials r_i so that $r_i(x, \ldots, x) = q_i(x)$. We thus need to verify that $r_i(\mu - \mu_T, \mu - \mu_T, \ldots, \mu - \mu_T)$ is small.

In order to handle multilinear polynomials, we treat them as a sequence of linear polynomials. We note that if $r(\mu, \mu, \ldots, \mu)$ is abnormally large, then so is $\mathbb{E}[r(G, G, \ldots, G)]$. This means that if we evaluate r at d random elements of T, we are relatively likely to get an abnormally large value, our goal is to find some linear polynomial L for which the distribution of $L(T)$ has enough discrepancies that we can filter T based on L. To do this, consider starting with $r(x_1, x_2, \ldots, x_d)$ where x_i are separate n-coordinate variables, and replacing the x_i one at a time with random elements of T. Since there is a decent probability that $r(t_1, \ldots, t_d)$ is large, it is reasonably likely that at some phase of this process, setting one of the variables causes the L^2-norm of r to jump by some substantial amount. In particular, there must be some settings of $t_1, \ldots, t_{a-1}$ so that for a random element t of T, we have that $r(t_1, \ldots, t_{a-1}, t, x_{a+1}, \ldots, x_d)$ will have substantially larger L^2-norm than $r(t_1, \ldots, t_{a-1}, x_a, x_{a+1}, \ldots, x_d)$ with non-negligible probability. We note that this would only rarely happen if t were distributed as $N(0, I)$, and this will allow us to filter.

To make this algorithm work, we note that

$$|r(t_1, \ldots, t_{a-1}, X, x_{a+1}, \ldots, x_d)|^2_{2, x_{a+1}, \ldots, x_d}$$

is a degree-2 polynomial with bounded trace-norm. Therefore, we need an algorithm so that if A is such a polynomial where $\mathbb{E}[A(T - \mu_T)]$ is large, we can produce a multifilter. This is done by writing A as an average of squares of linear polynomials. We thus note that there must be some linear polynomial L, where $\mathbb{E}[L(T - \mu_T)^2]$ is abnormally large. In particular, this implies that $L(T)$ and $L(G)$ have substantially different distributions, which should allow us to apply our basic multifilter. Also since L is degree-1, we have a priori bounds on $\mathrm{Var}[L(G)]$, which avoids the problem that has been plaguing us for much of this argument.

4 LEARNING SPHERICAL GAUSSIAN MIXTURE MODELS

In Section 4.1, we present a simpler learning algorithm that works when the components have the same covariance matrix. The general case of unknown (potentially different) covariances is more complex and is handled in Section 4.2. Section 4.3 contains our dimension-reduction procedures. In Section 4.4, we put everything together to obtain our final learning guarantees, including Theorem 1.4.

4.1 Learning Spherical GMMs: The Identity Covariance Case

We start by handling the important special case of this problem where each Gaussian component has identity covariance matrix. Note that our learning algorithm is robust to a small constant fraction of corrupted samples:

Proposition 4.1. *There is an algorithm that given a positive integer d, constants $1/2 > \alpha > 4\varepsilon \geq 0$, $0 < \tau < 1$, and sample access to a probability distribution $X = (1-\varepsilon)M + \varepsilon Y$, where $M = \sum w_i N(\mu_i, I)$ is a mixture of identity covariance Gaussians with $w_i \geq \alpha$ for all i, and so that $\|\mu_i - \mu_j\|_2$ is at least*

$$S \stackrel{\mathrm{def}}{=} C(\alpha^{-1/(2d)}\sqrt{d}(d+\log(1/\alpha))(\log(2+\log(1/\alpha)))^2 + \sqrt{\log(1/\varepsilon)})$$

for all $i \neq j$, takes $\mathrm{poly}((nd)^d \log(1/\tau)/(\varepsilon\alpha))$ samples from X, runs in time $\mathrm{poly}((nd\log(1/\tau)/\alpha)^{O(d)}, 1/\varepsilon)$, and, with probability at least $1-\tau$, returns a list of pairs (u_i, v_i), so that up to some permutation $|u_i - w_i| = O(\varepsilon)$ and $\|\mu_i - v_i\|_2 = \tilde{O}(\varepsilon/w_i)$.

Proof. The algorithm itself is very simple. We run our list-decoding algorithm to get a list of hypothesis means. We then associate each sample with the closest element of our list. We can then cluster points based on which means they are associated to and use this to learn the correct components. The algorithm is as follows:

Algorithm `LearnIdentityCovarianceGMM`
Input: Parameters $k, d \in \mathbb{Z}_+$, $\tau, \varepsilon > 0$ and sample access to X.

(1) Let T be a set of sufficiently many $\mathrm{poly}((nd)^d \log(1/\tau)/\alpha)$ samples from X.
(2) Run Algorithm `List-Decode-Gaussian` using T to obtain a list $H = (h_1, h_2, \ldots, h_m)$ with $m = O(1/\alpha)$.
(3) Let T' be a set of sufficiently many $\mathrm{poly}(nk/\varepsilon)$ samples from X.
(4) For each sample from T' associate it to the closest element of H in ℓ_2-distance.
(5) Let H' be the set of $h \in H$ so that at least a $2\alpha/3$-fraction of the elements of T' are associated to an element of H at most $S/10$ away from h.
(6) Define the relation on H' that $h \sim h'$ if and only if $\|h - h'\|_2 \le S/3$. If this does not define an equivalence relation on H' return "FAIL".
(7) For each equivalence class C of H', let T_C be the set of points in T' that are associated to elements of $C \subset H$.
(8) Let $u_C = |T_C|/|T'|$ for each C.
(9) For each C, run `Filter-Gaussian-Unknown-Mean` from [13] on T_C, and let v_C be the approximation of the mean obtained.
(10) Return the list of (u_C, v_C).

Note that for each i, X is simultaneously a mixture of $N(\mu_i, I)$ with weight α and some other distribution with weight $(1-\alpha)$. Therefore, for each i with probability at least $1 - \tau/(10k)$, there is some $h_j \in H$ with

$$\|h_i - \mu_i\|_2 = O(\alpha^{-1/(2d)}\sqrt{d}(d+\log(1/\alpha))(\log(2+\log(1/\alpha)))^2) \le S/100\,,$$

for C is sufficiently large. By a union bound, with probability at least $1 - \tau/10$, this occurs for every i. We assume this holds throughout the remainder of our analysis.

Let S_i be the set of elements of T' drawn from the component $N(\mu_i, I)$.

Lemma 4.2. *With probability $1 - \exp(-\Omega(S^2))$ over the samples from T', all but an $\exp(-\Omega(S^2))$-fraction of the elements of S_i are associated with elements $h \in H$ with $\|h - \mu_i\|_2 \le S/20$.*

Proof. The basic idea of the proof is the following: For any given $h \in H$ that is far from μ_i, there will be some $h' \in H$ that is much closer. A given sample point x will only be closer to h than h' if its projection to the line between them is more than half way there. However, this projection is distributed as a Gaussian, and therefore the probability that it is much larger than its mean is small.

It suffices to show that for each $h \in H$ with $\|h - \mu_i\|_2 > S/20$ the following holds: less than a $\exp(-\Omega(S^2))$-fraction of the elements of S_i are associated with h.

Firstly, assuming that the first step was successful, we know that there is an $h' \in H$ with $\|h' - \mu_i\|_2 < S/100$.

Let v be the unit vector in the direction of $h - h'$. We note that x is closer to h than h' if and only if $v \cdot x \ge v \cdot (h + h')/2$. However, we note that $v \cdot \mu_i \le v \cdot h' + S/100$, whereas, $v \cdot h = v \cdot h' + \|h - h'\|_2 \ge v \cdot h' + S/20$. The probability that $v \cdot X \ge \mathbb{E}[v \cdot X] + S/50$ for X drawn from $N(\mu_i, I)$ is $\exp(-\Omega(S^2))$. Thus, the probability that a sample drawn from $N(\mu_i, I)$ is closer to h than h' is $\exp(-\Omega(S^2))$.

Thus, by Markov's inequality, the probability that more than a $\exp(-\Omega(S^2))$-fraction of the elements of S_i are associated to h (with suitably small constant in the $\Omega(\cdot)$), is $\exp(-\Omega(S^2))$. Taking a union bound over h, does not change this asymptotic. □

Taking a union bound over i, we can assume that, with probability at least $1 - \tau/10$, we have that all but an $\exp(-\Omega(S^2))$ fraction of the points of S_i are associated with some h_j where $\|h_j - \mu_i\|_2 \le S/20$. In particular, this implies that every element of H within distance $S/20$ of some μ_i is in H'. Indeed, this holds for the following reason: With high probability, $|S_i| \ge (3\alpha/4) \cdot |T'|$ and at least $8/9$ fraction of elements in S_i are associated with h_j's that are within distance $S/20$ of μ_i. By the triangle inequality these h_j's are within distance $S/3$ of h. Conversely, any element of H not within distance $S/20$ of some μ_i has associated with it at most an $\varepsilon/10$-fraction of the elements of the union of the S_i's. This implies that with high probability less than $1.2\varepsilon < 2\alpha/3$ fraction of points in T are associated to *any* point of H not within distance $S/20$ of some μ_i. Therefore, all points of H' are within distance $3S/20$ of some μ_i, which implies that the relation on H' is an equivalence relation. Specifically, each equivalence class consists of the points in H' within distance $3S/20$ of some particular mean μ_i. Note in particular that this implies that there is exactly one equivalence class C for each μ_i.

Furthermore, Lemma 4.2 implies that all but an $\varepsilon/(10k)$-fraction of the samples from $N(\mu_i, I)$ are associated with elements of H in the class associated with μ_i. Furthermore, at most an ε-fraction of the other samples from T are associated to elements of this class. From this it immediately follows that $|u_i - w_i| \le 1.2\varepsilon$. Furthermore, the points associated with this class are an $O(\varepsilon/w_i)$-noisy version of $N(\mu_i, I)$. Therefore, `Filter-Gaussian-Unknown-Mean` returns a mean v_i with $\|v_i - \mu_i\|_2 = \tilde{O}(\varepsilon/w_i)$. This completes the proof of Proposition 4.1. □

4.2 Learning Spherical GMMs: The General Case

We now generalize the algorithm from the previous subsection to handle arbitrary mixtures of spherical Gaussians. When it is not the case that all of the covariance matrices are the same, things are substantially more complicated. We can recover a list H of candidate means only after first guessing the radius of the component that we are looking for. We can produce a large list of guesses and thereby obtain a list of hypotheses of size $\mathrm{poly}(n/\alpha)$. However, clustering becomes somewhat more difficult, as we do not know the radius at which to cluster. In particular, Steps 6 and 7 become difficult not knowing at what distance to stop considering two hypotheses part of the same cluster. This difficulty can be dealt with by doing a secondary test to determine whether or not the cluster that we have found contains many points at approximately the correct distance from each other.

Proposition 4.3. *There is an algorithm that, given a positive integer d, constants $1/2 > \alpha > 3\varepsilon \ge 0$, and sample access to a probability distribution $X = (1-\varepsilon)M + \varepsilon Y$, in dimension n larger than a sufficiently large multiple of $\log(\tau/\alpha)$, where $M = \sum_{i=1}^{k} w_i N(\mu_i, \sigma_i^2 I)$ is a mixture of spherical Gaussians with $w_i \ge \alpha$ for all i, and so that*

$\|\mu_i - \mu_j\|_2/(\sigma_i + \sigma_j)$ is at least

$$S \stackrel{\text{def}}{=} C(\alpha^{-1/(2d)}\sqrt{d}(d+\log(1/\alpha))(\log(2+\log(1/\alpha)))^2+\sqrt{\log(nk/\varepsilon)}) ,$$

for all $i \neq j$, where C is a sufficiently large universal constant, takes $\mathrm{poly}((dn)^d \log(1/\tau)/(\alpha\varepsilon))$ samples from X, runs in time

$$\mathrm{poly}((dn\log(1/\tau)/\alpha)^d/\varepsilon)$$

and, with probability at lest $1-\tau$, returns a list of triples (u_i, v_i, s_i), that satisfy the following conditions (up to some permutation): $|u_i - w_i| = O(\varepsilon)$, $\|\mu_i - v_i\|_2 = \tilde{O}(\varepsilon/w_i)\sigma_i$, and $|s_i - \sigma_i|/\sigma_i = \tilde{O}(\varepsilon/w_i)/\sqrt{n}$.

The detailed proof is given in the full version.

4.3 Dimension Reduction

In this section, we describe our dimension reduction scheme for the case of spherical mixtures. When the components have the same covariance, dimension reduction is quite simple and allows us to assume without loss of generality that the ambient dimension is $k-1$. The effect of dimension reduction for this case is that the runtime of the learning algorithm becomes somewhat better as a function of n.

When the components have arbitrary spherical covariances, we require a more complicated procedure that allows us to reduce the dimension down to $\mathrm{poly}(k/\varepsilon)$. In addition to improving the dependence on n in the runtime, this has the effect of removing the $\Omega(\sqrt{\log(n)})$ dependence in the separation condition of Proposition 4.3.

For the case of identity covariance components, we will require the following generalization of Theorem 4.2 of [41] or Corollary 3 of [50]:

Lemma 4.4. *Given $\varepsilon > 0$, suppose we take $\Omega(n\log(k/\tau)/(\varepsilon^4 w_{\min}^4))$ independent samples from $X = \sum_{i=1}^k w_i N(\mu_i, \sigma_i^2 I)$, where $w_i \geq w_{\min}$, and let W be the affine subspace of dimension $k-1$ containing the empirical mean $\widetilde{\mu}$ and spanned by the top $k-1$ eigenvectors of the empirical covariance $\widetilde{\Sigma}$. Then, with probability at least $1-\tau$, for all i, μ_i', the orthogonal projection of μ_i onto W, satisfies $\|\mu_i' - \mu_i\|_2 \leq \varepsilon\sigma$, where $\sigma^2 = \sum_i w_i\sigma_i^2$.*

Note that unlike Corollary 3 of [50], we only need W to be $k-1$ dimensional and unlike Theorem 4.2 of [41], we do not need the means μ_i to be bounded.

Proof. We first use standard facts about the empirical mean and covariance matrix of a single Gaussian:

Fact 4.5. *If we take $\Omega(n\log(1/\tau)/\varepsilon^2)$ independent samples from a Gaussian $N(\mu, \Sigma)$, then, except with probability τ, we have that the empirical covariance $\widetilde{\Sigma}$ and empirical mean $\widetilde{\mu}$ satisfy $(1-\varepsilon)\Sigma \preceq \widetilde{\Sigma} \preceq (1+\varepsilon)\Sigma$ and $(\widetilde{\mu}-\mu)^T\Sigma(\widetilde{\mu}-\mu) \leq \varepsilon^2$.*

Let $\delta = \varepsilon^2 w_{\min}/12$. By Chernoff bounds, the above fact and a union bound, we have that except with probability τ, since we have $\Omega(n\log(k/\tau)/(\delta^2 w_{\min}^2))$ samples, the fraction of samples from $N(\mu_i, \sigma_i^2 I)$, $\widetilde{w}_i$, satisfies $(1-\delta)w_i \leq \widetilde{w}_i \leq (1+\delta)w_i$, and the empirical covariance $\widetilde{\Sigma}_i$ and mean $\widetilde{\mu}_i$ of the samples coming from $N(\mu_i, \sigma_i^2 I)$ satisfy $(1-\delta)\sigma_i^2 I \preceq \widetilde{\Sigma}_i \preceq (1+\delta)\sigma_i^2 I$ and $\|\widetilde{\mu}_i - \mu_i\|_2 \leq \delta\sigma_i$. We assume that this holds.

Next note that we can write the empirical covariance as

$$\widetilde{\Sigma} = \sum_{i=1}^k \widetilde{w}_i \left(\widetilde{\Sigma}_i + (\widetilde{\mu}_i - \widetilde{\mu})(\widetilde{\mu}_i - \widetilde{\mu})^T\right) .$$

Since $\widetilde{\mu}$ is a convex combination $\widetilde{\mu} = \sum_{j=1}^k \widetilde{w}_j\widetilde{\mu}_j$ of the $\widetilde{\mu}_j$, the vectors $\widetilde{\mu}_i - \widetilde{\mu}$ span a $(k-1)$-dimensional subspace. For any unit vector v in the $(n-k+1)$-dimensional subspace orthogonal to this subspace, we have

$$v^T\widetilde{\Sigma}v = \sum_{i=1}^k \widetilde{w}_i v^T\widetilde{\Sigma}_i v \leq \sum_{i=1}^k (1+\delta)w_i(1+\delta)\sigma_i^2 \leq (1+3\delta)\sigma^2 .$$

Thus, the bottom $n-k+1$ eigenvalues of $\widetilde{\Sigma}$ are at most $(1+3\delta)\sigma^2$.

Now consider $\widetilde{\mu}_i'$, the orthogonal projection of $\widetilde{\mu}_i$ onto W. Let $v = (1/\|\widetilde{\mu}_i - \widetilde{\mu}_i'\|_2)(\widetilde{\mu}_i - \widetilde{\mu}_i')$. Since v is orthogonal to the top-k eigenvectors of $\widetilde{\Sigma}$, it follows that $v^T\widetilde{\Sigma}v \leq (1+3\delta)\sigma^2$. Since v is orthogonal to W which contains $\widetilde{\mu}_i'$ and $\widetilde{\mu}$, we have $v^T(\widetilde{\mu}_i' - \widetilde{\mu}) = 0$. Thus, we have

$$\begin{aligned}
(1+3\delta)\sigma^2 &\geq v^T\widetilde{\Sigma}v \\
&= \sum_{j=1}^k \widetilde{w}_j \left(v^T\widetilde{\Sigma}_j v + (v^T(\widetilde{\mu}_j - \widetilde{\mu}))^2\right) \\
&\geq (1-\delta)\sum_{j=1}^k w_j \left((1-\delta)\sigma_j^2 + (v^T(\widetilde{\mu}_j - \widetilde{\mu}))^2\right) \\
&\geq (1-2\delta)\sigma^2 + (1-\delta)w_i(v^T(\widetilde{\mu}_i - \widetilde{\mu}))^2 \\
&= (1-2\delta)\sigma^2 + (1-\delta)w_i(v^T(\widetilde{\mu}_i - \widetilde{\mu}_i') + v^T(\widetilde{\mu}_i' - \widetilde{\mu}))^2 \\
&= (1-2\delta)\sigma^2 + (1-\delta)w_i(\|\widetilde{\mu}_i - \widetilde{\mu}_i'\|_2 + 0)^2 .
\end{aligned}$$

Re-arranging, we have $\|\widetilde{\mu}_i - \widetilde{\mu}_i'\|_2 \leq \sqrt{5\delta\sigma^2/((1-\delta)w_i)}$. Setting $\delta = \varepsilon^2 w_{\min}/12$ gives $\|\widetilde{\mu}_i - \widetilde{\mu}_i'\|_2 \leq \varepsilon\sigma/2$.

Noting that projecting onto an orthogonal space reduces Euclidean distance, we have $\|\widetilde{\mu}_i' - \mu_i'\|_2 \leq \|\widetilde{\mu}_i - \mu_i\|_2$. The triangle inequality gives $\|\mu_i - \mu_i'\|_2 \leq \|\widetilde{\mu}_i - \mu_i\|_2 + \|\widetilde{\mu}_i - \widetilde{\mu}_i'\|_2 + \|\widetilde{\mu}_i' - \mu_i'\|_2 \leq \varepsilon\sigma/2 + 2\delta\sigma_i \leq \varepsilon\sigma/2 + \varepsilon\sqrt{w_i}\sigma_i/2 \leq \varepsilon\sigma$. This completes the proof. □

We now handle the general case:

Proposition 4.6. *Let $X = \sum_{i=1}^k w_i N(\mu_i, \sigma_i^2 I)$ be a k-mixture of spherical Gaussians in $\mathbb{R}^n$ with $w_i \geq \varepsilon$ for all i, for some $\varepsilon > 0$. There exists an algorithm that given k and ε, draws $\mathrm{poly}(nk/\varepsilon)$ samples from X, runs in sample-polynomial time, and returns an affine subspace W of dimension $\mathrm{poly}(k/\varepsilon)$, so that with high probability each μ_i is within distance $O(\varepsilon\sigma_i)$ of its projection onto W.*

Remark. Note that the above proposition can be combined with our algorithm from Section 4.2 by first finding W and then learning the projection of X onto W. Since we are now working in only $\mathrm{poly}(k/\varepsilon)$ dimensions, the latter does not require a $\log(n)$ dependence on S.

Proof. We start with the following observation: If we knew that all of the σ_i's were within a constant multiple of some known σ, we could simply scale X down by a factor of σ and then project onto the top k eigenvectors of the empirical covariance. This approach is used in [50]. The difficulty comes when the σ_i's are not close to each other. Doing this in this case would only give error $O(\varepsilon\sqrt{\sum_j w_j\sigma_j^2})$,

which will be larger than $O(\varepsilon\sigma_i)$ for some i. To deal with this issue, we notice that similar to the proof of Proposition 4.3, we can approximate the σ associated to a given sample by measuring how close it is to other samples. This will allow us to break our samples into several subsets each of which is a mixture of Gaussians with similar covariances.

It will also be important to note that our accuracy in measuring the radius of a Gaussian based on a few samples gets better as the dimension increases. Fortunately, we can assume without loss of generality that n is sufficiently large, as otherwise we can simply return $W = \mathbb{R}^n$. The dimension-reduction algorithm is as follows:

Algorithm `DimensionReduce`
Input: Parameters $k \in \mathbb{Z}_+$, $\varepsilon > 0$, and sample access to X.

(1) If n is not larger than a sufficiently large polynomial in k/ε, return $W = \mathbb{R}^n$.
(2) Let U be a set of $N = \mathrm{poly}(nk/\varepsilon)$ (for a sufficiently large polynomial) samples from X.
(3) For each $x \in U$, let $r(x) = \min_{y\in U, y\neq x} \|x-y\|_2/\sqrt{n}$.
(4) Define a relation $x \sim y$ if $r(x)$ and $r(y)$ are within a multiplicative factor of $(1\pm n^{-1/3})$. Let $\{C_j\}$ be the equivalence classes under the transitive closure of $\sim$.
(5) For each C_j:
  (a) let s_j be the minimum value of $r(x)$ for $x \in C_j$.
  (b) Compute $\widetilde{\mu}_j$ and $\widetilde{\Sigma}_j$, the empirical mean and covariance matrix of C_j.
  (c) Use PCA to find the $k-1$ eigenvectors $v_1, \ldots, v_{k-1}$ of $\widetilde{\Sigma}_j$ with the largest eigenvalues.
  (d) Let $W_j = \widetilde{\mu}_j + \mathrm{span} < v_1, \ldots, v_{k-1} >$.
(6) Return W, the affine span of the W_j's.

We note that we can assume that $n \gg \mathrm{poly}(N)$, for a sufficiently large polynomial or the algorithm trivially terminates in Step 1. We assume this throughout the rest of this proof.

In order to analyze the algorithm, we need to understand the distribution of the $r(x)$. To begin with, we note that:

Lemma 4.7. *With high probability over our samples, for every $x \neq y$ from U, with x drawn from $N(\mu_i, \sigma_i^2 I)$ and y drawn from $N(\mu_j, \sigma_j^2 I)$, we have that $\|x-y\|_2^2 = \left(\|\mu_i-\mu_j\|_2^2 + (\sigma_i^2+\sigma_j^2)n\right)(1+o(n^{-1/3}))$.*

Proof. We note that, for any given choice of i and j, a random pair of x and y satisfy this except with $\exp(-\Omega(n))$ probability. The lemma follows from a union bound over x and y. □

Taking a minimum, we find that:

Lemma 4.8. *With high probability, for all $x \in U$ drawn from $N(\mu_i, \sigma_i^2 I)$, we have that*

$$r(x) = \min_j \left(\sqrt{\|\mu_i-\mu_j\|_2^2/n + (\sigma_i^2+\sigma_j^2)}\right)(1+o(n^{-1/3})).$$

Proof. Assuming the conclusion of Lemma 4.7 holds, then the $r(x)$ is automatically at least this big and is at most this large assume that at least one (other) sample was drawn from $N(\mu_j, \sigma_j^2 I)$ for the minimizing j. This of course happens with high probability. □

Corollary 4.9. *With high probability, for all x drawn from $N(\mu_i, \sigma_i^2 I)$ we have that*

$$\sigma_i(1-o(n^{-1/3})) \leq r(x) \leq \sqrt{2}\sigma_i(1+o(n^{-1/3})) .$$

Proof. The lower bound is immediate. The upper bound follows from taking $j = i$. □

The above Corollary also has several other consequences. All of the x's coming from the same component will have $r(x)$ close to $\min_j(\sqrt{\|\mu_i-\mu_j\|_2^2/n + (\sigma_i^2+\sigma_j^2)})$ and thus all lie in the same C_j. This implies that there are at most k many classes C_j. Furthermore, since the x's coming from a single Gaussian component all have $r(x)$ within a $1+o(n^{-1/3})$ multiple of each other, it means that all of the $r(x)$, for $x \in C_j$, are within a $(1+n^{-1/3})^{O(k)}$ multiple of each other. Therefore, for each j, all of the $x \in C_j$ have $r(x)$ within a constant multiple of s_j. Thus, all of these x's come from Gaussians with σ_i within a constant multiple of s_j. Let S_j be the set of i such that all samples from $N(\mu_i, \sigma_i)$ are in C_j. By Lemma 4.4, the orthogonal projection μ_i' of μ_i for $i \in S_j$ onto W_j satisfies $\|\mu_i' - \mu_i\|_2 \leq \varepsilon\sigma$, where $\sigma^2 = \left(\sum_{i\in S_j} w_i\sigma_i^2\right) / \sum_{i\in S_j} w_i$. Since $\sigma = \Theta(s_j) = \Theta(\sigma_i)$ for each $i \in S_j$, we have that $\|\mu_i' - \mu_i\|_2 \leq O(\varepsilon\sigma_i)$. Therefore, since W contains W_j, μ_i is within $O(\varepsilon\sigma_i)$ of its projection onto W for all i.

Finally, since each W_j has dimension at most $k-1$ and since W is the sum of at most k of them, we have that $\dim(W) \leq k^2$. This completes the proof. □

4.4 Putting Everything Together

By combining Proposition 4.1 and Lemma 4.4, we immediately obtain the following corollary:

Corollary 4.10. *There is an algorithm that given a positive integer d, constants $1/2 > \alpha > 4\varepsilon \geq 0$, $0 < \tau < 1$, and sample access to a probability distribution $M = \sum w_i N(\mu_i, I)$ with $w_i \geq \alpha$ for all i, and so that $\|\mu_i - \mu_j\|_2$ is at least*

$$S \stackrel{\mathrm{def}}{=} C(\alpha^{-1/(2d)}\sqrt{d}(d+\log(1/\alpha))(\log(2+\log(1/\alpha)))^2 + \sqrt{\log(1/\varepsilon)})$$

for all $i \neq j$, takes $\mathrm{poly}\left(n(kd)^d \log(1/\tau)/(\varepsilon\alpha)\right)$ samples from X, runs in time $\mathrm{poly}\left(n(kd\log(1/\tau)/\alpha)^{O(d)}/\varepsilon\right)$, and, with probability at least $1-\tau$, returns a list of pairs (u_i, v_i), so that up to some permutation $|u_i - w_i| = O(\varepsilon)$ and $\|\mu_i - v_i\|_2 = \tilde{O}(\varepsilon/w_i)$.

Proof of Theorem 1.4. To prove this theorem, we will combine Proposition 4.6 with Proposition 4.3 and a few additional ingredients. In particular, running Proposition 4.6, we find a subspace W, as required. We note that the projection of X onto W is still a mixture of Gaussians with appropriate separations between the means to run Proposition 4.3. We note that because the dimension is now only $\mathrm{poly}(k/\delta)$, the $\log(n)$ term in S becomes a $\log(k/\delta)$, and the dependence on n in the sample complexity disappears. We can then learn w_i to error $O(\delta)$ and the projection of μ_i to W to error $\sigma_i O(\delta/w_i)$, which is within $\sigma_i O(\delta/w_i)$ of the true value of μ_i.

Learning approximations to the σ_i is slightly more difficult. Naively, we should only be able to learn it within error $\sigma_i O(\delta/w_i)/\sqrt{\dim(W)}$, which is not good enough. However, we note that samples from X can be reliably sorted (with δ probability of error) by which Gaussian they came from by considering which

equivalence class C from Proposition 4.3 the sample came from. Looking at the distances between pairs of the original samples in $\mathbb{R}^n$ whose projections end up in the same class, and taking the median, we can approximate σ_i to error $\sigma_i O(\delta/w_i)/\sqrt{n}$. This completes the proof. □

ACKNOWLEDGMENTS

I. D. is supported by NSF Award CCF-1652862 (CAREER) and a Sloan Research Fellowship. D. M. K. is supported by NSF Award CCF-1553288 (CAREER) and a Sloan Research Fellowship.

REFERENCES

[1] J. Acharya, I. Diakonikolas, J. Li, and L. Schmidt. 2017. Sample-Optimal Density Estimation in Nearly-Linear Time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*. 1278–1289.

[2] D. Achlioptas and F. McSherry. 2005. On Spectral Learning of Mixtures of Distributions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory (COLT)*. 458–469.

[3] J. Anderson, M. Belkin, N. Goyal, L. Rademacher, and J. R. Voss. 2014. The More, the Merrier: the Blessing of Dimensionality for Learning Large Gaussian Mixtures. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014*. 1135–1164.

[4] S. Arora and R. Kannan. 2001. Learning mixtures of arbitrary Gaussians. In *Proceedings of the 33rd Symposium on Theory of Computing*. 247–257.

[5] S. Balakrishnan, S. S. Du, J. Li, and A. Singh. 2017. Computationally Efficient Robust Sparse Estimation in High Dimensions. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*. 169–212.

[6] M.-F. Balcan, A. Blum, and S. Vempala. 2008. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. 671–680.

[7] M. Belkin and K. Sinha. 2010. Polynomial Learning of Distribution Families. In *FOCS*. 103–112.

[8] T. Bernholt. 2006. *Robust Estimators are Hard to Compute*. Technical Report. University of Dortmund, Germany.

[9] A. Bhaskara, M. Charikar, A. Moitra, and A. Vijayaraghavan. 2014. Smoothed analysis of tensor decompositions. In *Symposium on Theory of Computing, STOC 2014*. 594–603.

[10] S. C. Brubaker and S. Vempala. 2008. Isotropic PCA and Affine-Invariant Clustering. In *Proc. 49th IEEE Symposium on Foundations of Computer Science*. 551–560.

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

[12] S. Dasgupta. 1999. Learning mixtures of Gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. 634–644.

[13] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. 2016. Robust Estimators in High Dimensions without the Computational Intractability. In *Proceedings of FOCS'16*. 655–664.

[14] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. 2017. Being Robust (in High Dimensions) Can Be Practical. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*. 999–1008.

[15] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. 2017. Robustly Learning a Gaussian: Getting Optimal Error, Efficiently. *CoRR* abs/1704.03866 (2017). https://arxiv.org/abs/1704.03866 To appear in SODA'18.

[16] I. Diakonikolas, D. M. Kane, and A. Stewart. 2016. Robust Learning of Fixed-Structure Bayesian Networks. *CoRR* abs/1606.07384 (2016).

[17] I. Diakonikolas, D. M. Kane, and A. Stewart. 2016. Statistical Query Lower Bounds for Robust Estimation of High-dimensional Gaussians and Gaussian Mixtures. *CoRR* abs/1611.03473 (2016). http://arxiv.org/abs/1611.03473 In Proceedings of FOCS'17.

[18] I. Diakonikolas, D. M. Kane, and A. Stewart. 2017. Learning Geometric Concepts with Nasty Noise. *CoRR* abs/1707.01242 (2017). http://arxiv.org/abs/1707.01242

[19] I. Diakonikolas, D. M. Kane, and A. Stewart. 2017. List-Decodable Robust Mean Estimation and Learning Mixtures of Spherical Gaussians. *CoRR* abs/1711.07211 (2017). http://arxiv.org/abs/1711.07211

[20] J. Feldman, R. O'Donnell, and R. Servedio. 2006. PAC Learning Mixtures of Gaussians with No Separation Assumption. In *Proc. 19th Annual Conference on Learning Theory (COLT)*. 20–34.

[21] V. Feldman. 2017. A General Characterization of the Statistical Query Complexity. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*. 785–830.

[22] V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. 2013. Statistical algorithms and a lower bound for detecting planted cliques. In *Proceedings of STOC'13*. 655–664.

[23] V. Feldman, C. Guzman, and S. Vempala. 2017. Statistical Query Algorithms for Mean Vector Estimation and Stochastic Convex Optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '17)*. 1265–1277.

[24] V. Feldman, W. Perkins, and S. Vempala. 2015. On the Complexity of Random Satisfiability Problems with Planted Solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC, 2015*. 77–86.

[25] R. Ge, Q. Huang, and S. M. Kakade. 2015. Learning Mixtures of Gaussians in High Dimensions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*. 761–770.

[26] N. Goyal, S. Vempala, and Y. Xiao. 2014. Fourier PCA and robust tensor decomposition. In *Symposium on Theory of Computing, STOC 2014*. 584–593.

[27] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. 1986. *Robust statistics. The approach based on influence functions*. Wiley New York.

[28] M. Hardt and E. Price. 2015. Tight Bounds for Learning a Mixture of Two Gaussians. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*. 753–760.

[29] S. B. Hopkins and J. Li. 2017. Mixture Models, Robustness, and Sum of Squares Proofs. *CoRR* abs/1711.07454 (2017). arXiv:1711.07454 http://arxiv.org/abs/1711.07454 In STOC'18.

[30] D. Hsu and S. M. Kakade. 2013. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Innovations in Theoretical Computer Science, ITCS '13*. 11–20.

[31] P.J. Huber and E. M. Ronchetti. 2009. *Robust statistics*. Wiley New York.

[32] P. J. Huber. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics* 35, 1 (1964), 73–101.

[33] S. Janson. 1997. *Gaussian Hilbert Spaces*. Cambridge University Press, Cambridge, UK.

[34] R. Kannan, H. Salmasian, and S. Vempala. 2008. The Spectral Method for General Mixture Models. *SIAM J. Comput.* 38, 3 (2008), 1141–1156.

[35] M. Kearns. 1998. Efficient noise-tolerant Learning from statistical queries. *JACM* 45, 6 (1998), 983–1006.

[36] P. Kothari, J. Steinhardt, and D. Steurer. [n. d.]. Robust Moment Estimation and Improved Clustering via Sum of Squares. ([n. d.]). In STOC'18.

[37] K. A. Lai, A. B. Rao, and S. Vempala. 2016. Agnostic Estimation of Mean and Covariance. In *Proceedings of FOCS'16*.

[38] J. Li and L. Schmidt. 2017. Robust and Proper Learning for Mixtures of Gaussians via Systems of Polynomial Inequalities. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*. 1302–1382.

[39] A. Moitra and G. Valiant. 2010. Settling the polynomial learnability of mixtures of Gaussians. In *FOCS*. 93–102.

[40] K. Pearson. 1894. Contribution to the mathematical theory of evolution. *Phil. Trans. Roy. Soc. A* 185 (1894), 71–110.

[41] O. Regev and A. Vijjayraghavan. 2017. On Learning Mixtures of Well-Separated Gaussians. In *Proceedings of FOCS'17*. Full version available at https://arxiv.org/abs/1710.11592.

[42] P. J Rousseeuw and A. M Leroy. 2005. *Robust regression and outlier detection*. Vol. 589. John Wiley & Sons.

[43] J. Steinhardt, M. Charikar, and G. Valiant. 2017. Resilience: A Criterion for Learning in the Presence of Arbitrary Outliers. *CoRR* abs/1703.04940 (2017). http://arxiv.org/abs/1703.04940

[44] J. Steinhardt, P. W. Koh, and P. Liang. 2017. Certified Defenses for Data Poisoning Attacks. *CoRR* abs/1706.03691 (2017). http://arxiv.org/abs/1706.03691 To appear in NIPS 2017.

[45] J. Steinhardt, G. Valiant, and M. Charikar. 2016. Avoiding Imposters and Delinquents: Adversarial Crowdsourcing and Peer Prediction. In *NIPS*. 4439–4447.

[46] A. T. Suresh, A. Orlitsky, J. Acharya, and A. Jafarpour. 2014. Near-Optimal-Sample Estimators for Spherical Gaussian Mixtures. In *Advances in Neural Information Processing Systems (NIPS)*. 1395–1403.

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

[48] J.W. Tukey. 1975. Mathematics and picturing of data. In *Proceedings of ICM*, Vol. 6. 523–531.

[49] S. Vempala and G. Wang. 2002. A Spectral Algorithm for learning mixtures of distributions. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*. 113–122.

[50] S. Vempala and G. Wang. 2004. A spectral algorithm for learning mixture models. *J. Comput. System Sci.* 68, 4 (2004), 841 – 860.