Principled Approaches to Robust Machine Learning and Beyond

by

Jerry Zheng Li

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Author . . . .
Department of Electrical Engineering and Computer Science
August 31, 2018

Certified by. . . . .
Ankur Moitra
Rockwell International CD Associate Professor of Mathematics
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Principled Approaches to Robust Machine Learning and Beyond

by

Jerry Zheng Li

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

As we apply machine learning to more and more important tasks, it becomes increasingly important that these algorithms are robust to systematic, or worse, malicious, noise. Despite considerable interest, no efficient algorithms were known to be robust to such noise in high dimensional settings for some of the most fundamental statistical tasks for over sixty years of research.

In this thesis we devise two novel, but similarly inspired, algorithmic paradigms for estimation in high dimensions in the presence of a small number of adversarially added data points. Both algorithms are the first efficient algorithms which achieve (nearly) optimal error bounds for a number fundamental statistical tasks such as mean estimation and covariance estimation. The goal of this thesis is to present these two frameworks in a clean and unified manner.

We show that these insights also have applications for other problems in learning theory. Specifically, we show that these algorithms can be combined with the powerful Sum-of-Squares hierarchy to yield improvements for clustering high dimensional Gaussian mixture models, the first such improvement in over fifteen years of research. Going full circle, we show that Sum-of-Squares also can be used to improve error rates for robust mean estimation.

Not only are these algorithms of interest theoretically, but we demonstrate empirically that we can use these insights in practice to uncover patterns in high dimensional data that were previously masked by noise. Based on our algorithms, we give new implementations for robust PCA, new defenses for data poisoning attacks for stochastic optimization, and new defenses for watermarking attacks on deep nets. In all of these tasks, we demonstrate on both synthetic and real data sets that our performance is substantially better than the state-of-the-art, often able to detect most to all corruptions when previous methods could not reliably detect any.

Thesis Supervisor: Ankur Moitra
Title: Rockwell International CD Associate Professor of Mathematics

Acknowledgments

I would first like to thank my Ph.D advisor, Ankur Moitra, for his invaluable help and guidance throughout my Ph.D, both from a technical and more general perspective. I'd also like to thank my master's advisor Nir Shavit, for his help in orienting me in the research process, as well as being so supportive and willing to help me explore theoretical computer science. Additionally, I would like to thank my undergraduate advisors Dan Suciu, Paul Beame, and in particular, James Morrow, for introducing me to mathematics and theoretical computer science.

Of course, no researcher is an island, and I am also indebted to my wonderful collaborators and coauthors whom I've worked with along the way, including (but potentially not limited to) Jayadev Acharya, Dan Alistarh, Zeyuan Allen-Zhu, Paul Beame, Trevor Brown, Michael Cohen, Ilias Diakonikolas, Rati Gelashvili, Mohsen Ghaffari, Elena Grigorescu, Demjan Grubić, Chinmay Hegde, Sam Hopkins, Guatam Kamath, Daniel Kane, Kaan Kara, Justin Kopinsky, Ji Liu, Aleksander Mad̨ry, Ankur Moitra, Giorgi Nadiradze, Abhiram Natarajan, Krzysztof Onak, John Peebles, Sudeepa Roy, Ludwig Schmidt, Alistair Stewart, Jacob Steinhardt, Vikrant Singhal, Dan Suciu, Brandon Tran, Jonathan Ullman, and Ce Zhang. Without you all my results would have been far less complete and/or interesting.

I would also like to thank my many friends and colleagues at MIT and elsewhere who made my time here so memorable. Shoutouts to S. Achuuuuuuuuueour, Josh Alman, the entirety of BLAƆKPINK, Clement Cannononnnonẽ, Craig "My Name's Not Craig" Bodwin, Brynmor "Abra" Chapman, Aloni Cohen, Rati Gelashvili, Daniel Grier, Justin Holmgren, Sam Hopkins, Gauautam Kamath, Seulgi Kang, Sejeong "God" Kim, Albert "Sunghwan" Kwon, Rio LaVigne, "Grandpa" Will Leiserson, Quanquan Liu, Alex "Falco" Lombardi, Not Luke, Alex "Free Bird" Makelov, Dill Pickle McKay, Cam "Chris" Musco, Chris "Cam" Musco, Sam "The Spark" Park, John Peebles, Ilya Razenshteyn, "Lightning" Luke Schaeffer, Adam Sealfon, Aaron Sidford, Chaeyoung "smol" Son, Jennifer Tang, Brandon Tran, Alex Wein, Kai Xiao, and last and quite frankly least, Yang Yang "The Asiansoul" Zhao.

I also want to thank and remember Michael Cohen and Amelia Perry, two good friends and geniuses who are gone far before their time. Michael, thanks for all the fun times late at night in Stata shooting rockets, and for valiantly trying to teach me mirror prox. Amelia, thanks amongst other things, for taking the super cramped spot on the train on the way back from COLT. You were, in so many ways, a trooper.

Finally I'd like to thank my parents for supporting me.

Contents

List of Figures

List of Tables

A Note on the Content

This thesis presents a subset of results from the author's results during his Ph.D [GGLS14, AKLS15, ADH+15, LP15, ADLS16, DKK+16, ADLS17, LS17, BDLS17, AKLN17, ZLK+17, DKK+17, AGL+17, DGL+17, DKK+18a, HL18, ABK+18, DLS18, LMPS18, DKK+18b, AAZL18, KLSU18, TLM18], spanning roughly four (related) lines of work.

In the interest of presenting a somewhat coherent and concise thesis, it covers a single line of work, namely, robust estimation in high dimensions. It covers large parts of the papers [DKK+16, BDLS17, DKK+17, HL18, DKK+18b, TLM18]. Another paper in this line of work which we mostly omit is [DKK+18a], though we touch on parts of it.

The author also noted that in a number of theses he read, people would include quotes from their favorite pieces of poetry and/or classical literature. In part because the author is frankly too uncultured to know such material [Red15, Twi15], and in part for his own entertainment and happiness [Red14], the author decided to do something similar but instead with loose translations of lyrics from Korean pop songs [Gir07, Api16, Im17, IOI16, Red17a, BTS18b, IOI17, Lee17, IKO18]. The author hopes that the reader and his future self will forgive this frivolity.

Chapter 1

Introduction

Walking the many and unknowable paths,
I follow a dim light.
Let us do so together,
until the end,
into the new world.

1.1 A new perspective of robustness

The main question we seek to answer in this thesis is the following:

Question 1: *Given adversarially corrupted high dimensional data, how can we efficiently extract meaningful information from it?*

This question is purposefully left somewhat vague, and in this thesis we will explore a number of variations on this theme. In general, this question is of great interest to data scientists and computer scientists, both in theory and in practice. Classically this field has been known as *robust statistics*; more recently it has gained a lot of attention in machine learning as *adversarial ML*. This problem has been studied in both statistics and machine learning for over fifty years [Tuk60, Hub64], yet until recently the algorithmic aspects of this question were shockingly poorly understood.

In a 1997 retrospective on the development of robust statistics [Hub97], Peter Huber (one of the founders of the field), laments:

> “It is one thing to design a theoretical algorithm whose purpose is to prove [large fractions of corruptions can be tolerated] and quite another thing to design a practical version that can be used not merely on small, but also on medium sized regression problems, with a 2000 by 50 matrix or so. This last requirement would seem to exclude all of the recently proposed [techniques]."

The goal of this thesis is to answer Huber’s call to action and design estimators for a number of statistical and supervised learning tasks—including those from the original robust statistics papers—which are provably robust, and work in high-dimensions. Such estimators make the promise of robust learning in high dimension much closer to a reality.

The need for robustness in data analysis and machine learning is fairly universal. Systematic and uncontrolled noise can become part of a dataset in many, and often hard to avoid, ways. This noise can be due to model misspecification, since our simple models fail to capture all of the intricacies of the real world. It can be due to happenstance, if for instance small subpopulations of data are agglomerated into the large dataset. And it can be due to malicious adversaries, who wish to corrupt the algorithm’s performance. The latter in particular has become especially worrisome in the modern era of machine learning and data science, as we use these algorithms for increasingly important and sensitive applications.

To demonstrate the importance of robustness in modern data science and machine learning, let us briefly list a couple of examples where robustness plays a major role.

Feature extraction for biological data The first application is biological data (such as gene expression data). An important task in computational biology is to analyze and visualize genetic expression data. In doing so, systematic and uncontrolled noise can occur in many ways. For instance, labeling or measurement errors can create systematic outliers [RPW$^+$02, LAT$^+$08] that require painstaking manual effort to remove [PLJD10]. Additionally, since data is often amalgamated from many different labs, data sets for genetic expression are often contaminated with large amounts of systematic noise.

Moreover, even given clean data sets, because of the presence of small but genetically different subpopulations, general trends in genetic data can be obscured. For instance, as mentioned in [NJB$^+$08], the connection between genetic expression and geography they report can only be found after carefully pruning genetic information from immigrants. As a result, algorithms which are automatically robust to adversarial noise can help to speed up the process of discovery, or to find new patterns previously masked by these sources of noise.

Defending against data poisoning attacks The second motivation is machine learning security, where outliers can be introduced through *data poisoning* attacks [BNJT10] in which an adversary inserts malicious data into the training set. Recent work has shown that for high-dimensional datasets, even a small fraction of outliers can substantially degrade the learned model [BNL12, NPXNR14, KL17, SKL17]. This is especially worrisome in setting such as search engines or recommender systems, where it is natural to gather data from crowdsourced sources [KMY$^+$16]. However, when we do so, we can no longer fully trust our training dataset, and ignoring these security issues can have dangerous effects [BLA16]. For instance, there are instances where it is believed that search engines have been manipulated by a small group of malicious users injecting their own queries, potentially influencing important events like elections [ER15]. Clearly in such settings it would be ideal if these algorithms were resistant to such meddling.

One form of data poisoning attack we highlight in particular are *backdoor attacks.* Here, rather attemping to degrade the performance of the model on the test data, the goal is to implant a backdoor into the model, so that, given any test image, the adversary can perform some slight modifications to the image to have the model perform badly on this image. These attacks are harder to root out, since it is often even hard to detect whether or not the model has been backdoored. Such attacks have been discovered against deep neural networks. For instance, the authors of [GDGG17] were able to build a backdoor into neural nets used for stop sign detection in autonomous cars. By adding a small number of adversarially chosen points to the training set, they were able to cause the network to wrongly classify any image by adding their chosen perturbation. This attack is especially interesting—and dangerous—since the network behaves normally on unwatermarked test images.

Observe that in all of these applications, the data is often quite high dimensional. This is typical for most ML and data sciences applications nowadays. For instance, genetic data is often tens or hundreds of thousands of dimensions. However, as will be a recurring theme throughout this thesis, being robust in high dimensions can be quite challenging from an algorithmic perspective. Understanding the interplay between robustness and computation will be a large part of the contribution of this thesis.

1.1.1 Formalizing the question

We approach Question 1 from a mathematical perspective, so the first order of business is to rigorously specify what we mean in Question 1. In general, there are 3 components to making this question formal:

- **What kind of assumptions do we make on the data?** In this thesis, we will primarily make distributional assumptions on the inliers. That is, we assume the uncorrupted points are drawn i.i.d. from some "nice" distribution. However, it will often be the case that we will give deterministic conditions on the inliers under which our algorithms work.

- **What kinds of corruptions are considered?** In this work we focus on a strong notion of corruption, namely *gross* corruption, where we assume that a small fraction of samples are completely corrupted. There are of course a number of different models of corruption, such as statistical notions such as model misspecification, additive models such as Hüber's contamination model, etc., however, in large part, our model subsumes these notions. That is, in our model, the adversary corrupting the samples is allowed to do more than the adversary in these other models of corruption. Despite this, we achieve strong statistical guarantees against this adversary, much stronger than were even known for the weaker models of corruption considered previously. In Section 1.4 we discuss these notions of corruption and compare them in more detail.

- **What sort of information do we wish to recover?** Depending on the problem, it is natural to ask to recover different types of information. The most basic questions center around recovering information about the distribution of uncorrupted points. However, we can also consider questions in a supervised model. For instance, we could ask to learn some model as if the data were uncorrupted.

1.1.2 Overview of the problems

In this thesis, we build a theory of how to approach Question 1, starting from the most fundamental questions, and working our way to more general and complicated settings. Even for the most basic questions, prior to our work, the picture was quite incomplete. While these simple setups are quite specialized, by building on these ideas, we are then able to tackle increasingly general problems.

In all these settings, there are three important criteria we are concerned with:

- **Statistical error** Given enough (possibly corrupted) samples, we wish to obtain small error. Observe that, unlike in traditional minimax theory, in general we cannot expect this error to go to zero as we take more and more samples, simply because as we get more samples, we also receive more corrupted data, since we assume that a constant fraction of the data points may be corrupted. As our data is often extremely high dimensional, it is important that our error guarantees are *dimension independent*. This is also intimately related to classical notions studied in robust statistics such as *breakdown point*.

- **Runtime** Naturally, we wish to be able to actually run our algorithms, and thus it is important that they are efficient computationally. Traditionally in learning theory this has meant polynomial time, but since datasets nowadays are very large, ideally we want extremely fast algorithms. As we shall discuss below, previous approaches to robust estimation that worked in high dimensions had runtimes which were *exponential* in the parameters. We are not only able to obtain the first polynomial time robust estimators in high dimensions, but in fact our algorithms are fast enough to be run on large, high dimensional data sets in practice!

- **Sample Complexity** We want our error guarantees to kick in even when we do not have too many samples. While this is of course a very important measure, it turns out that our algorithms tend to naturally be (nearly) sample optimal, and so throughout this thesis we will generally emphasize this point less.

With these points in mind, we can now state the problems we consider in our thesis.

The starting point: robustly learning a Gaussian Here, we assume that we are given samples $X_1, \ldots, X_n \in \mathbb{R}^d$ which are drawn i.i.d. from a distribution D which is a d-dimensional Gaussian with mean μ and covariance Σ, except an ε-fraction of these points have been arbitrarily changed. The goal is then to recover the underlying Gaussian. Since a Gaussian is determined by its mean and covariance, this question is equivalent to learning μ, Σ under the appropriate metrics. In this setting, we are able to obtain the first polynomial time estimators which achieve nearly optimal rates in the presence of adversarial noise:

Theorem 1.1.1 (informal, see Theorems 2.2.1, 5.3.1, 5.4.1)**.** *Fix $\varepsilon > 0$. Let $X_1, \ldots, X_n$ be a sufficiently large set of samples from an unknown Gaussian G, except an ε-fraction of them are arbitrarily corrupted. There is a polynomial time algorithm which, given $X_1, \ldots, X_n$, outputs $\widehat{G}$ so that with high probability, the total variation distance between G and $\widehat{G}$ is at most $O(\varepsilon \log 1/\varepsilon)$.*

We remark that it is easily demonstrated that $\Omega(\varepsilon)$ error is necessary for *any* algorithm, given any number of samples. This is in contrast to traditional minimax settings, where we expect the error to go to zero as the number of samples goes to infinity. This is because even though we get to take more samples, the adversary always gets to corrupt an ε-fraction of them. Thus, our error is nearly optimal, up to log factors.

Moreover, the algorithms we design also have other very nice properties. In particular, they have nearly optimal sample complexities as well: their sample complexities match (up to logarithmic factors) the optimal sample complexity for the non-robust version of the problem. Thus, in this regard, robustness comes "for free" in this setting.

Robust parameter estimation While the above problem is arguably the most basic setting, it ends up being somewhat brittle, since the algorithms may implicitly use the Gaussianity of the inliers. In practice, data may not have such nice concentration, or may not exactly have the nice moment structure that Gaussians possess. From this perspective, a natural generalization of the above problem is then to only assume that the distribution of inliers D is still "nice" in some sense, in that it has bounded moments up to some degree. The goal then is not to learn D (since without stronger assumptions this is impossible), but to recover stastistics of D such as the mean or covariance.

We observe that while these problems are mathematically quite simple to state, already they are of great interest in practice. For instance, the problem of robustly estimating the covariance is intimately related to robust principal component analysis (PCA), since the principal components of any dataset are simply the top eigenvectors of the covariance. Hence, if we can robustly recover the covariance, we can simply read off the top eigenvectors, and perform robust PCA. Prior methods for robust PCA worked under orthogonal or weaker conditions on either the data matrix or the corruptions. As a result, our methods are able to detect patterns on real high dimensional data sets that these previous methods could not find.

The Gaussian learning algorithms that achieve 1.1.1 go through this recipe, and give algorithms for robust mean and covariance estimation of a Gaussian. As a result, ingredients from these algorithms can immediately give some results for robust parameter estimation. In particular, they give nearly optimal results for mean estimation of sub-Gaussian distributions with known covariance:[1]

Theorem 1.1.2 (informal, see Theorems 2.2.13 and 5.3.1)**.** *Fix $\varepsilon > 0$. Let $X_1, \ldots, X_n$ be a sufficiently large set of samples from an unknown sub-Gaussian distribution D with mean μ and covariance I, except an ε-fraction of them are arbitrarily corrupted. There is a polynomial time algorithm which, given $X_1, \ldots, X_n$, outputs $\widehat{\mu}$ so that with high probability,*

$$\|\mu - \widehat{\mu}\|_2 = O(\varepsilon \sqrt{\log 1/\varepsilon}) .$$

As before, this error is optimal up to log factors, and our algorithms are sample optimal up to log factors. Moreover, for this important subproblem, we give algorithms which are extremely efficient: only requiring at most $\widetilde{O}(d)$ passes through the data.[2] Thus not only do we achieve polynomial time estimators, but we are able to obtain practical estimators.

However, sometimes data is not so well concentrated. In such settings, is it possible to recover robust parameter recovery guarantees? We also give results for robust mean estimation under much weaker assumptions. Specifically, we are able to show that non-trivial robust estimation is possible even when we only assume bounded second moments:

Theorem 1.1.3 (informal, see Theorem 5.5.11)**.** *Fix $\varepsilon > 0$. Let $X_1, \ldots, X_n$ be a sufficiently large set of samples from an unknown distribution D with mean μ and bounded covariance, except an ε-fraction of them are arbitrarily corrupted. There is a polynomial time algorithm which, given $X_1, \ldots, X_n$, outputs $\widehat{\mu}$ so that with high probability,*

$$\|\mu - \widehat{\mu}\|_2 = O(\sqrt{\varepsilon}) .$$

[1] As we will define more formally in Section 1.4, a distribution is sub-Gaussian if it concentrates "at least as well" as a Gaussian along every univariate projection.

[2] Throughout this thesis we let $\widetilde{O}(f) = O(f \log^{O(1)} f)$.

It turns out that with these weaker assumptions, this weaker error guarantee is optimal up to constants, and as before, our algorithm is sample optimal up to log factors. This setting will prove extremely crucial for some of the later applications in the thesis.

Given that we now have tight results for sub-Gaussian distributions (where in a sense all moments are controlled), and results for distributions with only bounded second moments, it is a natural question to ask if we can interpolate between these two extremes. By using much more complicated algorithmic techniques, we are able to partially resolve this question:

Theorem 1.1.4 (informal, see Theorem 4.6.1)**.** *Fix $\varepsilon > 0$ sufficiently small and let $t \geq 4$. Let D be a distribution over $\mathbb{R}^d$ with mean μ so that its ℓth moments are bounded by those of a Gaussian, and this bound is given by a "simple certificate", for all $\ell \leq t$. Let $X_1, \ldots, X_n$ be a set of samples from D, where an ε-fraction of these have been arbitrarily corrupted. If $n = \Omega\left(d^{O(t)}(1/\varepsilon)^{O(1)}\right)$, there is an algorithm which takes $X_1, \ldots, X_n$ with running time $(d^t\varepsilon)^{O(t)}$ and outputs $\widehat{\mu}$ so that with high probability*

$$\|\widehat{\mu} - \mu\|_2 \leq O(\varepsilon^{1-1/t}) \ .$$

This algorithm is based on the powerful Sum-of-squares (SoS) hierarchy which will also play an important role for the next result. We remark that while the criterion that the bound have a "simple certificate" is a rather technical one, it can be shown (see e.g. [KS18]) that this applies to almost all classically studied distributions.

Clustering well-separated Gaussian mixture models The next problem is on the surface unrelated to robustness, however, it shares some deep and interesting technical connections to robust estimation, and in particular, Theorem 1.1.4. A (uniform) mixture of k distributions $D_1, \ldots, D_k$ is the distribution where samples are generated via the following process: first, draw i uniformly from $\{1, \ldots, k\}$, then output a sample from D_i. Mixture models, and especially, mixtures of Gaussians, are pervasive in practice. For instance, any statistic of a heterogenous population consisting of a number of separate sub-populations with different distributions of this statistic is well-modeled by a mixture model.

A well-studied and important question in this setting is to cluster data points drawn from a Gaussian mixture model. That is, given samples from a mixture over k isotropic Gaussians, recover with low error which component each sample came from. Without additional assumptions this is information theoretically impossible (for instance, consider the setting where all the Gaussians are identical). Thus, to make this problem well-formed, we impose separation conditions on the Gaussians, that is, the means of the Gaussians are located far apart from one another. The main question in this setting is what sort of separation is necessary to efficiently cluster the samples. Until our work, there was an exponential gap between what was known information theoretically and what could be handled using efficient algorithms: one can show that $\Omega(\sqrt{\log k})$ separation suffices to cluster the points with high probability, however, prior efficient algorithms required separation at least $\Omega(k^{1/4})$.

By leveraging algorithmic connections to robust statistics, we are able to drastically improve this:

Theorem 1.1.5 (Informal, see Theorem 4.5.1)**.** *For every $\gamma > 0$ there is an algorithm with running time $(dk)^{O(1/\gamma^2)}$ using at most $n \leq k^{O(1)} d^{O(1/\gamma)}$ samples which, given samples $x_1, \ldots, x_n$ from a uniform mixture of k spherical Gaussians in d dimensions with means $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ satisfying $\|\mu_i - \mu_j\|_2 \geq k^\gamma$ for each $i \neq j$, returns estimators $\hat{\mu}_1, \ldots, \hat{\mu}_k \in \mathbb{R}^d$ such that $\|\hat{\mu}_i - \mu_i\|_2 \leq 1/\operatorname{poly}(k)$ with high probability.*

We remark that our algorithms can also be generalized to work for non-uniform mixtures, and also to mixtures of distributions of the type described in Theorem 1.1.4.

Defending backdoor attacks on deep networks A related and recently discovered security threat to machine learning algorithms, and specifically deep neural networks, are known as *backdoor attacks.* Here, as in data poisoning attacks, an adversary injects a small fraction of adversarially chosen points into the data set. However, the goal is not to lower test accuracy, as it was before. Rather, the goal is to exploit the overparameterized nature of deep networks to build a “backdoor” into the network. The goal of the adversary is to cause the network to perform normally on normal test images. However, the adversary should be able to slightly alter any test image with their chosen perturbation, and cause the image to be misclassified. These attacks are especially insidious, since it can be hard to detect whether or not a network has been compromised or not, as its behavior looks normal on test images.

Prior to our work, no candidate defenses were known for this problem. We demonstrate empirically that algorithms based on the methods presented in this thesis are able to defeat all known backdoor attacks, in our experiments on CIFAR-10. Intuitively, it seems that known backdoor attacks cause a shift in the distribution of the learned representation, which our methods are able to detect. As a result, we are able to consistently remove almost all of the poisoned data points, causing the attack to fail. Due to the poorly understood nature of deep networks, we are not able to provide rigorous guarantees for our algorithm, but we view it as an important first step towards principled defenses to such attacks.

Robust stochastic optimization The previous problems were all in the regime of unsupervised learning. Another important setting is supervised learning, where we are given labeled data points, and the goal is to perform regression, or to learn a classifier for these data points.

As above, a natural question is whether we can perform supervised learning (i.e. get low test error) when a small fraction of adversarially chosen data points and/or labels are injected into the data set. In adversarial machine learning, such attacks are known as *data poisoning* attacks.

To unify this setting, we observe that many of these problems fall under the umbrella of *stochastic optimization*. Here, there is some distribution D over functions f, and the goal is to minimize $\overline{f}(x) = \mathbb{E}_{f \sim D}[f(x)]$. This setting is extremely general, and subsumes a large fraction of important supervised learning algorithms, including least-squares and ridge regression, logistic regression, support vector machines, etc.

We show that given black box access to an algorithm for a stochastic optimization task, it is possible to "robust-ify" it with minimal overhead, so that it is guaranteed to achieve good error, even in the presence of adversarial noise:

Theorem 1.1.6 (informal, see Theorem 7.2.1)**.** *Let D be a "nice" distribution over functions f, and let $\overline{f}(x) = \mathbb{E}_{f \sim D}[f(x)]$. Suppose we have black-box access to an algorithm $\mathcal{A}$ which, given $f_1, \ldots, f_n \in \mathrm{supp}(D)$, finds an approximate minimizer for $\frac{1}{n} \sum_{i=1}^{n} f_i(x)$. Then, for any $\varepsilon > 0$, there is an efficient algorithm which, given $\mathcal{A}$ and samples from D, where an ε-fraction of these samples may be arbitrarily corrupted, finds an approximate minimizer for $\overline{f}$ with error at most $O(\sqrt{\varepsilon})$.*

As a result, this gives us a general "meta-framework" for solving a number of optimization problems in the presence of data poisoning attacks, with provable guarantees. We then verify on synthetic and real data that our defenses achieve state-of-the-art accuracy against state-of-the-art attacks for two of these problems, namely, ridge regression and SVM.

1.2 Main contributions

In this section, we describe our contributions in this thesis to this area in more detail.

1.2.1 Overview

Rather than focusing on each problem in turn, it will be convenient for us to introduce two frameworks for solving the problem of robustly learning a Gaussian which achieve nearly identical statistical guarantees. These two frameworks can then be extended in different ways to attack different subsets of the problems described above.

Unknown Convex Programming Our first approach is based on the principles of convex programming. Developing this approach and its applications will be the main subject of Chapters 2-4.

We show that robustly learning parameters of a Gaussian can be written as a convex, but unknown, minimization problem. This convex program seeks to assign weights to the samples corresponding to how much they can be trusted to give good estimates of the true parameters. While the objective is unknown, we show that we can devise an approximate separation oracle for this minimization problem, allowing us to optimize this objective. This algorithm appeared in the following paper:

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

We give two applications of this approach. Our first is to robust sparse estimation, which is described in Chapter 3. In many settings, the parameters of interest are not completely arbitrary, but rather are quite structured, and in particular, sparse. For instance, when analyzing genetic data, most genes do not matter, and thus the set of important coordinates is often quite small. When there is no adversarial noise, it has been shown that we can capitalize on this to substantially improve statistical performance. We show that, by adapting the unknown convex programming approach, we can recover many of the same statistical guarantees, in the presence of adversarial noise. Interestingly, along the way we uncover some new candidate statistical-computational tradeoffs that seem to only arise in the presence of noise. This is based on the following paper:

- Sivaraman Balakrishnan, Simon S. Du, Jerry Li, Aarti Singh, *Computationally efficient robust sparse estimation in high dimensions*, appeared in COLT 2017 [BDLS17], which was a merger of two independent preprints [Li17, DBS17]. In this thesis we will focus on presenting the results in [Li17].

Our second application, described in Chapter 4, is to robust parameter estimation and high dimensional clustering. We show that by reinterpreting the convex program in the Sum of Squares (SoS) hierachy, we are able to generalize and lift the program to take into consideration higher order moment information. We can then use this formulation to substantially improve the separation needed to cluster Gaussian mixture models. Recall that to cluster a mixture of k Gaussians in high dimensions, previous efficient algorithms required separation $\Omega(k^{1/4})$, whereas $\Omega(\sqrt{\log k})$ separation suffices information theoretically. We give polynomial time algorithms which can tolerate separation $\Omega(k^{\varepsilon})$ for any constant $\varepsilon > 0$, and we are able to recover the information theoretic threshold in quasi-polynomial time. Interestingly, this same framework can also be used to improve robust parameter estimation for a wide class of sub-gaussian distributions. This is based on the following paper:

- Samuel B. Hopkins and Jerry Li, *Mixture Models, Robustness, and Sum of Squares Proofs*, appeared in STOC 2018 [HL18].

Filtering Our second approach, which we call *filtering*, is based on iteratively rejection sampling. Developing this approach and its applications will be the main subject of Chapters 5-7.

While the unknown convex programming approach assigns soft weights to each sample point corresponding to how much it believes that the sample is good, filtering assigns each point a score depending on how much it believes it is corrupted, and removes the ones with scores above a threshold, and then repeats the process. We show that when the good points come from a Gaussian, by carefully choosing how to assign these scores and the threshold, we can guarantee that we always remove more bad points than good points. As a result, we show that this algorithm always makes progress.

We also generalize this approach to work for distributions with bounded second moment. Moreover, filtering is quite practical; its pseudocode is quite simple, and each iteration of filtering runs in nearly-linear time. We have implemented this algorithm, and have demonstrated that it significantly improves upon the performance of previous algorithms on both synthetic and real data sets. Filtering was first introduced in the same paper as unknown convex programming [DKK^{+}16], and the generalizations and experiments are based on:

- Ilias Diakonikolas, Guatam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart, *Being Robust (in high dimensions) can be Practical*, appeared in ICML 2017 [DKK^{+}17].

In general filtering appears to hold much promise in real world applications. In Chapter 6, we demonstrate the utility of filtering on synthetic and real data sets.

As a concrete application on real data, we demonstrate that our method is able to recover patterns in high dimensional genomic data, even in the presence of adversarial noise, even when previous methods fail. This is based on results which first appeared in [DKK$^+$17].

Additionally, in the same chapter, we show that these ideas have applications to defending watermarking attacks against deep networks. We show that for known backdoor attacks, by mapping the data points to their learned representations resulting from a neural network, spectral methods can distinguish between the true data points from the watermarked points. As a result, they can be easily detected and removed. This section is based on the following paper:

- Brandon Tran, Jerry Li, Aleksander Mądry, *Spectral Signatures in Backdoor Attacks for Neural Networks*, in submission to NIPS 2018 [TLM18].

Finally, in Chapter 7, we dramatically generalize the filter to give results for for robust stochastic optimization. We show that insights developed from filtering can be combined with any black-box optimizer as a defense against data poisoning attacks. Our framework enjoys theoretical worst-case error guarantees, and also improves upon the error achieved by state of the art defenses against data poisoning attacks for ridge regression and SVM. This section is based on the following paper:

- Ilias Diakonikolas, Guatam Kamath, Daniel M. Kane, Jerry Li, Jacob Steinhardt, Alistair Stewart, *Sever: A Robust Meta-Algorithm for Stochastic Optimization*, in submission to NIPS 2018 [DKK$^+$18b].

1.3 A recipe for efficient robust estimation

In this section, we describe the basic geometry which is at the heart of the algorithmic ideas developed in this thesis. These ideas yield a very general recipe for algorithms for a large class of robust estimation tasks. At a very high level, the key insight is the following:

Corruptions to the empirical moment of a data set necessarily leave detectable traces in the spectrum of higher moments.

Our algorithms will crucially exploit this structural property. If, for instance, the goal is to detect whether the mean (the first moment) has been corrupted, then the key information can often be found in the empirical second moment (or even higher moments, in some cases). If alternatively, the goal is to detect whether the covariance (the second moment) has been corrupted, then we should look at the fourth moment, etc.

This method turns out to be quite powerful, and quite general. In particular, this works even when we have only quite weak assumptions on the structure of the unknown distribution. As a result, this meta-algorithm has found application in a number of settings, a subset of which we will cover in this thesis.

The remainder of this section is dedicated to informally justifying this recipe. Before we do so, it will be informative to understand the challenges that efficient robust estimation in high dimensions face.

1.3.1 Why is robust estimation in high dimensions hard?

Why was so little known about efficient robust estimation in high dimensions? Let us consider the simplest setting, to understand the main conceptual difficulties. Namely, let us consider the setting where we get $X_1, \ldots, X_n$ from a Gaussian in d dimensions with identity covariance, and mean μ, where an ε-fraction of these samples are arbitrarily corrupted. This is (essentially) the problem initially considered in the seminal work of [Tuk60, Hub64] that introduced robust statistics—yet until very recently no efficient algorithms were able to achieve the right error rate for this problem! Without understanding this problem, it will be difficult to attack the more complicated settings, so it is definitely worth spending some time here.

The state of affairs for this problem prior to our work can be summarized briefly in Table 1.1. In short, all known algorithms for this basic problem fit into one of two categories.

Dimensionality	Error guarantee	Efficient?
In low dimensions		
Median [folklore]	$\Theta(\varepsilon)$	**Yes**
Pruning [folklore]	$\Theta(\varepsilon\sqrt{\log 1/\varepsilon})$	**Yes**
In d dimensions		
Tukey Median [Tuk60]	$\Theta(\varepsilon)$	**No**
Geometric Median [Web29]	$\Theta(\varepsilon\sqrt{d})$	**Yes**
Tournament [folklore, see e.g. Ch. 6 of [DL12]]	$\Theta(\varepsilon)$	**No**
Pruning [folklore]	$\Omega(\varepsilon\sqrt{d})$	**Yes**
Coordinatewise median [folklore]	$\Theta(\varepsilon\sqrt{d})$	**Yes**
RANSAC [FB87], many iterations	$\widetilde{O}(\varepsilon)$	**No**
RANSAC [FB87], few iterations	$\Omega(\infty)$	**Yes**
Our results [DKK+16]	$O(\varepsilon\sqrt{\log 1/\varepsilon})$	**Yes**

Table 1.1: Overview of the known results for robustly learning the mean of a Gaussian prior to our work. Green indicates that the algorithm achieves the qualitatively desirable behavior for the given attribute, and red [Red17b] indicates that it does not.

- **Computationally intractable** The algorithm would require time which was exponential in the number of dimensions and/or samples, or would require solving a computational problem which is NP-hard in the worst case.

- **Statistically suboptimal** The error guarantees of the algorithm would provably degrade as the dimensionality of the data increased. Generally, the error would grow *polynomially* with the dimension. As we are interested in extremely high dimensional tasks, this renders the output of the algorithm uninformative.

Thus, as Huber lamented, the techniques developed for robust statistics were limited in applicability to the regime of roughly 10 to 50 dimensions. This is a far cry from modern day settings.

The barrier at $\Omega(\varepsilon\sqrt{d})$ So why is the problem so hard for efficient algorithms? In particular, why does it seem that efficient algorithms get stuck at $\Omega(\varepsilon\sqrt{d})$?

Let us consider a representative efficient algorithm, and see why this algorithm gets stuck. As it turns out, the same basic problem is at the heart of the issue for most, if not all, previously proposed efficient algorithms.

Let us consider the pruning algorithm. This algorithm is very basic: it simply attempts to remove all points which are "obviously" too far to be from the true distribution, and hopes that it has removed them all. It then takes the empirical mean of the remaining points in the data set.

This algorithm actually works pretty well in low dimensions (see Table 1.1). Intuitively, all points from a Gaussian in low dimension will be quite close to the true mean. Thus, if the outliers wish to survive the pruning, they must also be quite close to true mean, at distance roughly $O(\sqrt{\log 1/\varepsilon})$. Since there are an ε-fraction of outliers, they cannot corrupt the value of the empirical mean by more than $\widetilde{O}(\varepsilon)$. This is demonstrated pictorially in the figure on the left in Figure 1-1.

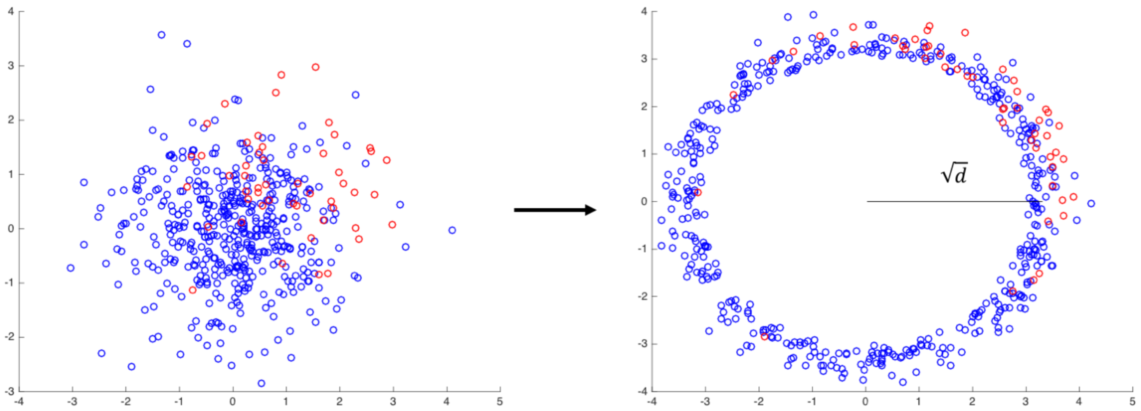


Figure 1-1: The qualitative difference between low dimensional and high dimensional robust estimation. Blue points are inliers and red points are outliers. On the left: the behavior of data in low dimensions. On the right: the behavior of data in high dimensions.

However, in high dimensions, this begins to degrade badly. This is because in high dimensions, we expect a typical sample from a Gaussian to have norm $\Theta(\sqrt{d})$. Thus, as depicted in the picture on the right in Figure 1-1, in high dimensions we should really think of samples from a Gaussian as living on a shell of radius roughly $\Theta(\sqrt{d})$. As a result, given a point in the shell, the algorithm cannot reliably distinguish if it is a outlier or not. Thus, if all the outliers also live within this shell, they will survive the pruning. As a result, there are an ε-fraction of corruptions, each of which can contribute a $\Omega(\sqrt{d})$ to the error. This results in an error of $\Omega(\varepsilon\sqrt{d})$.

Informally, what we've argued is that any method which attempts to determine whether a sample is an outlier at an individual sample level must get stuck at $\Omega(\varepsilon\sqrt{d})$. To surpass this barrier, we must somehow look at more *global* information of the corruptions.

1.3.2 Breaking the $O(\varepsilon\sqrt{d})$ barrier: spectral signatures

We now explain how the frameworks proposed in this thesis circumvent this difficulty. While the two frameworks may look somewhat different algorithmically, they are based on a shared information theoretic intuition. A major goal of this thesis is to flesh out this connection. Indeed, many of the applications we give for one approach or the other in this thesis can be achieved using the other approach, though with certain caveats both ways.

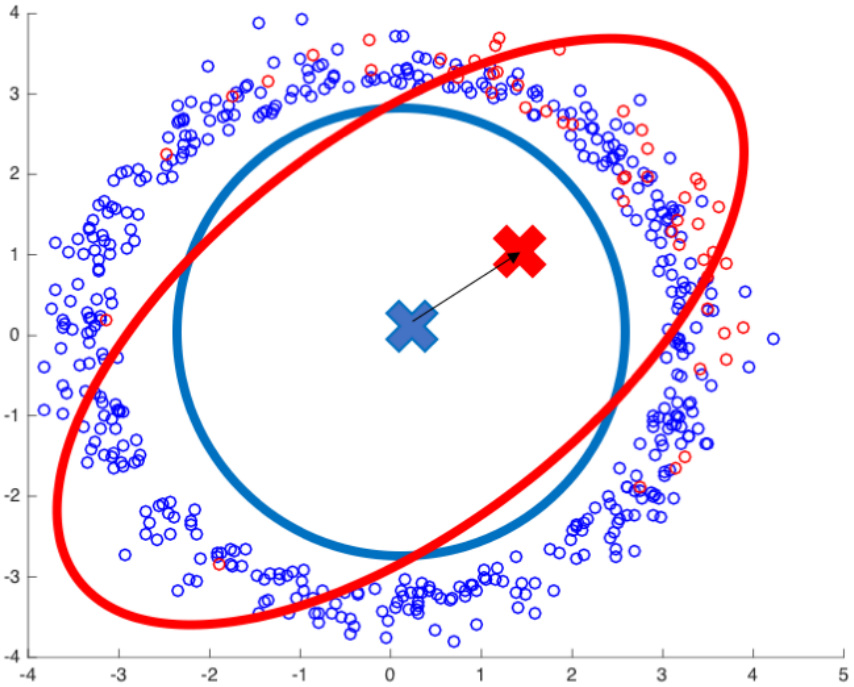


Figure 1-2: A cartoon to explain the phenomena of spectral signatures.

For the case of mean estimation, we provide a cartoon of this intuition in Figure 1-2. As before, blue points are inliers, drawn from an isotropic distribution. Red points are outliers, designed to change the empirical mean while blending in with the inliers. The blue X denotes the true mean of the distribution. Let us denote this μ. The red X denotes the empirical mean of the corrupted dataset, which we call $\widehat{\mu}$. The rough idea is as follows: we do not wish to identify which points are individually outliers. Indeed, as we argued above, to do so would incur an error which would necessarily grow polynomially with the dimension. Instead, we only care if the corrupted points work together in aggregate to change the empirical mean of the data set.

We next ask how this can happen. Certainly the mean of the uncorrupted points will concentrate nicely to the true mean of the distribution. Thus, if the mean $\widehat{\mu}$ of the whole dataset is far from the true distribution, this means that along the direction $\mu - \widehat{\mu}$, the corrupted points must be the source of the deviation, as we see in Figure 1-2. Since there are comparatively fewer corrupted points than there are inliers, this can only happen if the corrupted points are actually quite far out in this direction.

In fact, they must be so far out that this causes the variance of the total dataset in this direction to be noticeably larger than it should be. In Figure 1-2, the blue circle denotes the true covariance of the distribution, which is a circle, since the distribution is isotropic. However, the empirical distribution, the red oval, clearly has a large component in the direction roughly corresponding to $\mu - \widehat{\mu}$. As a result, this direction can be detected as a large eigenvector of the empirical covariance. This phenomena is something we call a *spectral signature*, and is a specific instance of a more general behavior: to detect deviations in a empirical moment caused by a small number of adversarially corrupted points, it suffices to consider spectral properties of higher moments.

1.4 Notation and preliminaries

For any natural number n, we let $[n] = \{1, \dots, n\}$. Throughout this thesis, we will always let n denote the number of samples we have taken.

For any $r > 0$ and any $\mu \in \mathbb{R}^d$, we let $B(\mu, r) = \{x \in \mathbb{R}^d : \|x - \mu\|_2 < r\}$ be the ℓ_2-ball of radius r around μ. For any two vectors $u, v \in \mathbb{R}^d$, we let $\langle u, v \rangle = \sum_{i=1}^d u_i v_i$ denote their usual inner product, and we let $\|v\|_2 = \langle v, v \rangle^{1/2}$ denote the ℓ_2 norm of v. It is easily verified that ℓ_2 is self-dual, i.e.

$$\|v\|_2 = \sup_{\|u\|_2 = 1} \langle u, v \rangle \; .$$

For any matrix $M \in \mathbb{R}^{m \times d}$ with singular values $\sigma_1 \geq \sigma_2 \ldots \sigma_r \geq 0$, where $r = \max(m, d)$, we let $\|M\|_2 = \sigma_1$ be its spectral norm, and $\|M\|_F = \left(\sum_{i=1}^r \sigma_i^2\right)^{1/2}$ denote its Frobenius norm. In a slight abuse of notation, given two matrices $A, B \in \mathbb{R}^{m \times d}$, we let $\langle A, B \rangle = \mathrm{tr}(A^\top B)$ to be the inner product between these two matrices.

When $M \in \mathbb{R}^{d \times d}$ is a Hermitian matrix, it can be verified that

$$\|M\|_2 = \sup_{\|u\|_2 = 1} \langle u, Mu \rangle \; , \qquad \|M\|_F = \sup_{\|A\|_F = 1} \langle A, M \rangle \; .$$

We let $\succeq$ denote the Loebner PSD ordering on symmetric matrices. Given $\Sigma \in \mathbb{R}^{d \times d}$ with $\Sigma \succ 0$, we let $\Sigma^{1/2}$ denote its matrix square root, and for any matrix $M \in \mathbb{R}^{d \times d}$ we let

$$\|M\|_\Sigma = \left\| \Sigma^{-1/2} M \Sigma^{-1/2} \right\|_F$$

denote the *Mahalanobis norm induced by* Σ of M. One can check that this value is invariant under choice of matrix square root, i.e. for any A so that $A^\top A = \Sigma^{-1}$, we have $\|M\|_\Sigma = \|AMA\|_F$.

Given k distributions $F_1, \ldots, F_k$ with PDFs $f_1, \ldots, f_k$ and mixing weights $w_1, \ldots, w_k$ so that $\sum_{i=1}^k w_i = 1$ and $w_i \geq 0$ for all $i \in [k]$, we let the mixture of $F_1, \ldots, F_k$ with mixing weights $w_1, \ldots, w_k$, denoted $D = \sum_{i=1}^k w_i F_i$, be the distribution with PDF $\sum_{i=1}^k w_i f_i$. This corresponds to the distribution whose samples are generated via the following process: first, choose i from $[k]$ with probability w_i, then output an independent sample from F_i. Given a sample $X \sim D$, we say i is its corresponding component if it was drawn from F_i. If the mixing weights are uniform, we say that the mixture is a uniform mixture.

For convenience throughout this thesis, we will often conflate probability distributions and their probability density functions (PDFs). Hopefully the context makes it clear which one we are discussing at any given time. When the usage may be ambiguous we will clarify.

1.4.1 The Gaussian distribution, sub-gaussian distributions

A univariate Gaussian (also known as a normal distribution) is specified by a mean $\mu \in \mathbb{R}$ and a variance $\sigma^2 > 0$, is denoted $\mathcal{N}(\mu, \sigma^2)$, and has PDF given by

$$\mathcal{N}(\mu, \sigma^2)(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\sigma^2}{2}(x-\mu)^2\right) .$$

When $\mu = 0$ and $\sigma^2 = 1$, we say that the distribution is the *standard normal* distribution.

Amongst the many important properties of Gaussians are their concentration properties. For instance, an important concentration property we will require is the following:

Fact 1.4.1. *Let G be the standard normal distribution. For any $T > 0$, we have*

$$\Pr_{X \sim G}\left[|X| \geq T\right] \leq \exp(-T^2/2) .$$

To generalize this to beyond Gaussians, we say that a univariate distribution D is *sub-gaussian* with variance proxy s^2 if

$$\mathop{\mathbb{E}}_{X \sim D}\left[\left(X - \mathop{\mathbb{E}}_{X \sim D}[X]\right)^k\right] \leq \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,s^2)}\left[X^k\right] ,$$

for all k even. It can be shown (see e.g. [RH17]) that this implies (indeed, is equivalent to the fact) that any sub-gaussian distribution has the same concentration properties as a Gaussian with the same variance.

A multivariate Gaussian distribution the natural generalization of a Gaussian to high dimensions. It is specified by a mean vector $\mu \in \mathbb{R}^d$ and a covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ so that $\Sigma \succ 0$, is denoted by $\mathcal{N}(\mu, \Sigma)$, and has probability distribution function given by

$$\mathcal{N}(\mu, \Sigma)(x) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) .$$

The PDF of an example Gaussian is pictured in Figure 1-3. When $\Sigma = \sigma^2 I$ for some $\sigma > 0$, we say that the corresponding Gaussian is *spherical*, and when $\Sigma = I$, we say that the Gaussian is *isotropic*.

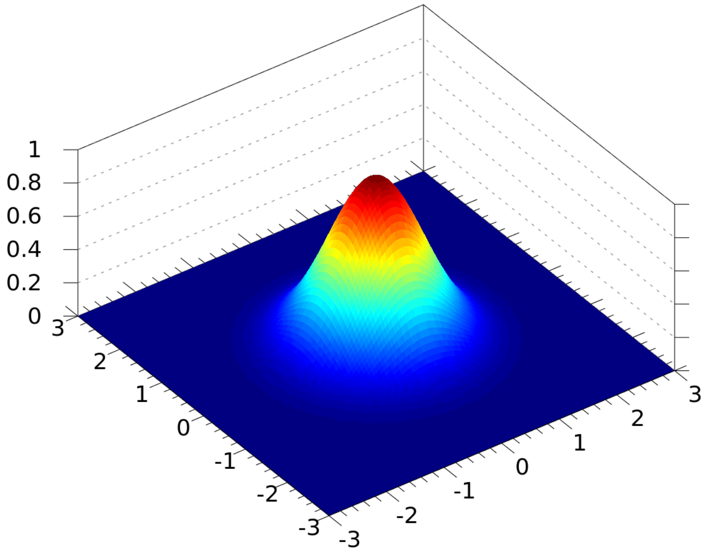


Figure 1-3: The PDF of a 2-dimensional Gaussian.

Finally, to generalize sub-gaussianity to multivariate settings, we say that a distribution D is *sub-gaussian with variance proxy* Σ if for all unit vectors $u \in \mathbb{R}^d$, the distribution $\langle u, X \rangle$ where $X \sim D$ is sub-gaussian with variance proxy $u^\top \Sigma u$. That is, the distribution looks sub-gaussian along all one dimensional projections. In analogy with multivariate Gaussians, we say a sub-gaussian distribution is *isotropic* if its covariance (which is always guaranteed to exist) is the identity.

1.4.2 Distances and divergences between distributions

Total variation distance Given two distributions F, G over a shared probability space Ω with PDFs f, g respectively, we define the *total variation distance* between F and G (also known as the *statistical distance*), denoted $d_{\mathrm{TV}}(F, G)$, to be

$$d_{\mathrm{TV}}(F, G) = \sup_A \left(\Pr_{X \sim F}[X \in A] - \Pr_{X \sim G}[X \in A] \right) = \frac{1}{2} \int_\Omega |f(x) - g(x)| \, .$$

This is a natural and well-studied metric of similarity between distributions. It is especially well suited for our setting because it measures how well samples from F, G can be coupled. Recall a *coupling* between two distributions F, G is a distribution over $\Omega \times \Omega$ whose marginals are distributed as F and G, respectively. Then, the following fact is well-known:

Fact 1.4.2 (folklore, see e.g. [Dur10]). *For any two distributions F, G, we have*

$$d_{\mathrm{TV}}(F, G) = \sup_{(X,Y)} \Pr\left[X \neq Y\right] \, ,$$

where the supremum is taken over all couplings of F and G.

In other words, given two distributions F and G with total variation distance ε, it is possible to transform samples from F to samples from G by changing at most an ε-fraction of the samples on average. Vice versa, if it is possible to change only an ε-fraction of samples of F to mimic G perfectly, then their total variation distance is at most ε. Since we are interested in constant amounts of gross corruptions, this naturally lends itself to the study of recovery in total variation distance. Moreover, as we shall see shortly, learning a Gaussian in total variation distance corresponds to recovering the parameters of the Gaussian in the natural affine invariant manner. Indeed, when the two Gaussians are isotropic, we have that learning them in TV distance is equivalent up to a constant factor to learning the means in ℓ_2:

Fact 1.4.3 (folklore, see e.g. [DKK+18a], Lemma 1). *Let $\varepsilon > 0$ be sufficiently small.*

Let $\mu_1, \mu_2 \in \mathbb{R}^d$ so that $\|\mu_1 - \mu_2\|_2 = \varepsilon$. Then, we have

$$d_{\mathrm{TV}}(\mathcal{N}(\mu_1, I), \mathcal{N}(\mu_2, I)) = \left(\frac{1}{\sqrt{2\pi}} + o(1) \right) \varepsilon \; .$$

For completeness of exposition we include the proof of this fact in Appendix A.1.

KL divergence Another important measure of similarity between distributions is the *Kullbeck-Liebler divergence* or KL divergence, also known as *relative entropy*. Given F, G with PDFs f, g respectively, the KL divergence between them, denoted by $d_{\mathrm{KL}}(F \| G)$, is given by

$$d_{\mathrm{KL}}(F \| G) = \int_\Omega \log f(x) \log \frac{f(x)}{g(x)} dx \; .$$

While KL divergence is not a metric (it is asymmetric), and while we will not directly study KL divergence, it is a very useful tool for us in the study of recovery in TV distance. This is because of a couple of reasons. First, we have the following classical inequality, which allows us to relate KL divergence to TV distance:

Fact 1.4.4 (Pinsker's inequality, see e.g. [CT06])**.** *Given two probability distributions F, G, we have*

$$d_{\mathrm{TV}}(F, G) \leq \sqrt{\frac{1}{2} d_{\mathrm{KL}}(F \| G)} \; .$$

The second reason is that there is a very convenient closed form formula for the KL divergence between two multivariate Gaussians:

Fact 1.4.5 (folklore)**.** *Let $\mu_1, \mu_2 \in \mathbb{R}^d$ and let $\Sigma_1, \Sigma_2 \in \mathbb{R}^{d \times d}$ be positive definite. Let $\mathcal{N}_1 = \mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}_2 = \mathcal{N}(\mu_2, \Sigma_2)$. Then we have*

$$d_{\mathrm{KL}}\left(\mathcal{N}_1 \,\|\, \mathcal{N}_2\right) = \frac{1}{2} \left(\mathrm{tr}(\Sigma_2^{-1} \Sigma_1) + (\mu_2 - \mu_1)^\top \Sigma_2^{-1} (\mu_1 - \mu_2) - k + \log \left(\frac{\det(\Sigma_2)}{\det(\Sigma_1)} \right) \right) \; . \tag{1.1}$$

Combining these two facts allows us to bound the TV distance between two Gaussians with the same mean by the difference in their covariances in the Mahalanoubis norm induced by either:

Corollary 1.4.6 (folklore, see e.g. [DKK+16], Corollary 2.14)**.** *Fix $\varepsilon > 0$ sufficiently small. Let Σ_1, Σ_2 be so that $\|\Sigma_1 - \Sigma_2\|_{\Sigma_2} = \varepsilon$. Then $d_{\mathrm{TV}}(\mathcal{N}(0, \Sigma_1), \mathcal{N}(\Sigma_2)) \leq O(\varepsilon)$.*

For completeness we provide a proof of Corollary 1.4.6 in Appendix A.1. We remark that it is not hard to see that in the regime where ε as defined in Corollary 1.4.6 is small, this bound is tight up to constant factors, though we will not require this. We also remark that the same technique can recover Fact 1.4.3, albeit losing some constant factors along the way. Still, KL divergence and Pinsker's inequality allow us to exactly characterize the TV distance (up to constant factors) between multivariate Gaussians.

1.4.3 Dealing with tensors

Let $\otimes$ denote the Kronecker product on matrices. We will make crucial use of the following definition:

Definition 1.4.1. For any matrix $M \in \mathbb{R}^{d \times d}$, let $M^\flat \in \mathbb{R}^{d^2}$ denote its canonical flattening into a vector in $\mathbb{R}^{d^2}$, and for any vector $v \in \mathbb{R}^{d^2}$, let $v^\sharp$ denote the unique matrix $M \in \mathbb{R}^{d \times d}$ such that $M^\flat = v$.

We will also require the following definitions:

Definition 1.4.2. Let $\mathcal{S}_{\mathrm{sym}} = \{M^\flat \in \mathbb{R}^{d^2} : M \text{ is symmetric}\}$, let $\mathcal{S} \subseteq \mathcal{S}_{\mathrm{sym}}$ be the subspace given by

$$\mathcal{S} = \{v \in \mathcal{S}_{\mathrm{sym}} : \mathrm{tr}(v^\sharp) = 0\} ,$$

and let Π_S and $\Pi_{\mathcal{S}^\perp}$ denote the projection operators onto $\mathcal{S}$ and $\mathcal{S}^\perp$ respectively. Finally let

$$\|v\|_{\mathcal{S}} = \|\Pi_{\mathcal{S}} v\|_2 \text{ and } \|v\|_{\mathcal{S}^\perp} = \|\Pi_{\mathcal{S}^\perp} v\|_2 .$$

Moreover, for any $M \in \mathbb{R}^{d^2 \times d^2}$, let

$$\|M\|_{\mathcal{S}} = \sup_{v \in \mathcal{S} - \{0\}} \frac{v^T M v}{\|v\|_2^2} \, .$$

In fact, the projection of $v = M^\flat$ onto $\mathcal{S}$ where M is symmetric can be written out explicitly. Namely, it is given by

$$M = \left(M - \frac{\mathrm{tr}(M)}{d} I \right) + \frac{\mathrm{tr}(M)}{d} I \, .$$

By construction the flattening of the first term is in $\mathcal{S}$ and the flattening of the second term is in $\mathcal{S}^\perp$. The expression above immediately implies that $\|v\|_{\mathcal{S}^\perp} = \frac{|\mathrm{tr}(M)|}{\sqrt{d}}$.

1.4.4 Types of adversarial noise

Here we formally define the types of adversarial corruption we study throughout this thesis. Throughout this thesis, we will always let ε denote the fraction of corrupted points, and we will always take it to be a sufficiently small constant. All of our algorithms will for for $\varepsilon \in (0, c]$ for some universal constant c sufficiently small. The largest c for which our algorithms work is known as the *breakdown point* of the algorithm, and is a well-studied object in robust statistics. However, in this thesis we will not focus on optimizing for breakdown point.

Recall that n will always denote the number of (potentially corrupted) samples we have. For convenience, we will always assume that εn is an integer value; by either slightly increasing ε and/or n by a small constant this can always be ensured.

The first, and most powerful, model we consider, will be that of adversarial corruption:

Definition 1.4.3 (ε-corruption)**.** Fix $\varepsilon \in (0, 1/2)$, and let D be a distribution. We say that a dataset $X_1, \ldots, X_n$ is a *ε-corrupted set of samples from D* if it is generated via the following process:

- n samples $Y_1, \ldots, Y_n$ are drawn i.i.d. from D.

- A computationally unbounded adversary inspects $Y_1, \ldots, Y_n$, arbitrarily alters an ε-fraction of these, then returns the altered set of samples in any arbitrary order.

Given an ε-corrupted set of samples $X_1, \ldots, X_n$, we let $S_{\mathrm{bad}} \subset [n]$ denote the set of indices of corrupted samples, and we let $S_{\mathrm{good}} = [n] \setminus S_{\mathrm{bad}}$.

Observe that while the uncorrupted points are originally independent, because the adversary is allowed to remove an ε-fraction of them after inspecting them, the points in S_{good} *may be dependent*. Getting around this dependency will be a crucial part of obtaining good error guarantees.[3]

All of the results presented in this thesis will hold for this strong notion of corruption. However, for completeness, we also mention other classically considered notions of robustness. The first is the notion of a *oblivious adversary*:

Definition 1.4.4 (Oblivious ε-corruption)**.** Fix $\varepsilon \in (0, 1/2)$, and let D be any distribution. We say that $X_1, \ldots, X_n$ is a *obliviously ε-corrupted set of samples from D* if they are drawn i.i.d. from some distribution D' with $d_{\mathrm{TV}}(D, D') \leq \varepsilon$.

This notion of corruption is also known as *model misspecification* in statistics. Perhaps it is not clear *a priori* that oblivious ε-corruption is weaker than ε-corruption, but this follows from the following lemma:

Lemma 1.4.7. *Fix $\varepsilon, \delta > 0$, and let D be a distribution. Let $X_1, \ldots, X_n$ be an obliviously ε-corrupted set of samples from D. Then, with probability $1 - \delta$, it is an $\left[\left(1 + O\left(\sqrt{\frac{\log 1/\delta}{n}} \right) \right) \cdot \varepsilon \right]$-corrupted set of samples from D.*

Proof. By definition, there exists a distribution D' be so that $X_1, \ldots, X_n$ are drawn i.i.d. from D', and so that $d_{\mathrm{TV}}(D, D') \leq \varepsilon$. By Fact 1.4.2, these samples can be coupled to $X'_1, \ldots, X'_n$ drawn i.i.d. from D so that for each i, $X_i = X'_i$ with probability $1 - \varepsilon$. Thus, the adversary simply uses these X'_i, and outputs X_i only when the

[3] Note that technically all of these sets should technically be considered multisets, as samples may be duplicated, especially in the corrupted sets. However, this does not meaningfully affect anything, and so for simplicity of notation, throughout this paper we will simply refer to these as sets, and use set notation and operations.

coupling disagrees. By a Chernoff bound, with probability $1-\delta$, the number of indices which disagree is $\left[\left(1+O\left(\sqrt{\frac{\log 1/\delta}{n}}\right)\right)\cdot\varepsilon\right]n$. $\square$

Thus, with a subconstant loss in the number of corrupted samples, we can simulate obliviously corrupted samples by corrupted samples. Up to this loss, observe that the adaptive corruption model is strictly stronger than the oblivious corruption model. This difference does not appear to be very meaningful, but it will be useful for us to think of adaptive corruption when analyzing our algorithms.

The last kinds of corruptions are strictly weaker, but have been studied fairly extensively, and are still of interest:

Definition 1.4.5 (ε-additively corruption)**.** Fix $\varepsilon \in (0,1/2)$, and let D be a distribution. We say a set of samples $X_1,\ldots,X_n$ is an *ε-additively corrupted set of samples from D* if it is generated via the following process:

- $(1-\varepsilon)n$ samples $Y_1,\ldots,Y_{(1-\varepsilon)n}$ are drawn i.i.d. from D.
- A computationally unbounded adversary inspects $Y_1,\ldots,Y_{(1-\varepsilon)n}$, adds εn arbitrarily chosen points to the data set, and returns the result in any arbitrary order.

As before, we let S_{bad} denote the set of corrupted samples, and S_{good} denote the remaining set of samples.

There is an analogous definition of oblivious additive corruption:

Definition 1.4.6 (oblivious ε-additive corruption)**.** Fix $\varepsilon \in (0,1/2)$. We say $X_1,\ldots,X_n$ is an *obliviously ε-additively corrupted set of samples from D* if they are drawn i.i.d. from $D' = (1-\varepsilon)D+\varepsilon F$, where F is an arbitrary distribution.

As an historical aside, we note that this was the model of corruption considered in Huber's original paper [Hub64], and is also known as *Huber's contamination model.*

As in the general case, the oblivious additive adversary can be simulated at a sub-constant loss by the adaptive additive adversary. The main difference in these additive settings when compared to the general corruption setting is that the adversary cannot remove good points; that is, the good points remain i.i.d. from the original distribution.

1.4.5 Robustly learning a Gaussian

With this we can now finally formally state the problem of robustly learning a Gaussian. This will be the first problem we solve with both unknown convex programming and filtering, and will serve as the launching point to all the other problems we consider in this thesis.

Problem 1.4.1 (Robustly learning a Gaussian)**.** Fix $\varepsilon > 0$, and let $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$ be symmetric and positive definite. Given an ε-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$, output $\widehat{\mu}$ and $\widehat{\Sigma}$ minimizing $d_{\mathrm{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\widehat{\mu}, \widehat{\Sigma}))$.

This naturally decomposes into two parametric sub-problems:

Problem 1.4.2 (Robust estimation of location)**.** Fix $\varepsilon > 0$, and let $\mu \in \mathbb{R}^d$. Given an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$, output $\widehat{\mu}$ minimizing $\|\mu - \widehat{\mu}\|_2$.

Problem 1.4.3 (Robust estimation of scale)**.** Fix $\varepsilon > 0$, and let $\Sigma \in \mathbb{R}^{d \times d}$ be symmetric and postive definite. Given an ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$, output $\widehat{\Sigma}$ minimizing $\|\Sigma - \widehat{\Sigma}\|_\Sigma$.

As stated above, solving Problem 1.4.1 (and its corresponding sub-problems Problem 1.4.2 and Problem 1.4.3) will prove to be an important starting point for understanding a number of other problems down the line.

Robust parameter recovery A more general problem is to ask for the same sorts of parameter recovery guarantees as in Problem 1.4.2, but for other classes of distributions.

Problem 1.4.4 (Robust mean estimation)**.** Fix $\varepsilon > 0$, and a class of distributions $\mathcal{D}$ over $\mathbb{R}^d$. Given an ε-corrupted set of samples from some $D \in \mathcal{D}$, output $\widehat{\mu}$ minimizing $\|\mu - \widehat{\mu}\|_2$.

The types of guarantees that can be achieved for this problem of course depend on $\mathcal{D}$. We remark that robust covariance estimation (up to maybe a constant factor loss) can also be thought of as a special case of this problem. Finally, we can also consider other norms (see e.g. [SCV18]), but in this thesis we will mostly restrict our attention to ℓ_2, although we touch on the geometry involved with robust mean estimation in other norms in Chapter 3.

1.4.6 Prior work

Before we dive into our results, we pause here to recap what was known about Problem 1.4.1 before the initial dissemination of [DKK$^+$16], the basis of the most fundamental results presented in our thesis. There has been a flurry of work concurrent to and subsequent to this by the author as well as many other researchers; for the sake of narrative we will present these with the corresponding sections of the thesis.

Our results fit in the framework of *density estimation* and *parameter learning* which are both classical problems in statistics with a rich history (see e.g., [BBBB72, DG85, Sil86, Sco92, DL12]). While these problems have been studied for several decades by different communities, the computational complexity of learning is still not well understood, even for some surprisingly simple distribution families. Most textbook estimators are hard to compute in general, especially in high-dimensional settings. In the past few decades, a rich body of work within theoretical computer science has focused on designing computationally efficient distribution learning algorithms. In a seminal work, Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie [KMR$^+$94] initiated a systematic investigation of the computational complexity of distribution learning. Since then, efficient learning algorithms have been developed for a wide range of distributions in both low and high-dimensions [Das99, FM99, AK01, VW02, CGG02, MR05, BV08, KMV10, MV10, BS10a, DDS12, CDSS13, DDO$^+$13, CDSS14a, CDSS14b, HP15a, DDS15, DDKT16, DKS16b, DKS16a, ADLS17].

Our general question of robust learning also resembles learning in the presence of malicious errors [Val85, KL93]. There, an algorithm is given samples from a distribution along with their labels according to an unknown target function. The adversary is allowed to corrupt an ε-fraction of both the samples and their labels. A sequence of works studied the problem of learning a halfspace with malicious noise in the setting where the underlying distribution is a Gaussian [Ser03, KLS09], culminating in the work of Awasthi, Balcan, and Long [ABL17], who gave an efficient algorithm that finds a halfspace with agreement $O(\varepsilon)$. There is no direct connection between their problem and ours, especially since one is a supervised learning problem and the other is unsupervised. However, we note that there is an interesting technical parallel in that the works [KLS09, ABL17] also uses spectral methods to detect outliers. Both their work and our algorithm for agnostically learning the mean are based on the intuition that an adversary can only substantially bias the empirical mean if the corruptions are correlated along some direction. Our other algorithms are also based on spectral techniques but need to handle many significant conceptual and technical complications that arise when working with higher moments or binary product distributions.

Another connection is to the work on robust principal component analysis (PCA). PCA is a transformation that (among other things) is often justified as being able to find the affine transformation $Y = \Sigma^{-1/2}(X - \mu)$ that would place a collection of Gaussian random variables in isotropic position. One can think of our results on agnostically learning a Gaussian as a type of robust PCA that tolerates gross corruptions, where entire samples are corrupted. This is different than other variants of the problem where random sets of coordinates of the points are corrupted [CLMW11], or where the uncorrupted points were assumed to lie in a low-dimensional subspace to begin with [ZL14, LMTZ15]. Finally, Brubaker [Bru09] studied the problem of clustering samples from a *well-separated* mixture of Gaussians in the presence of adversarial noise. The goal of [Bru09] was to separate the Gaussian components from each other, while the adversarial points are allowed to end up in any of clusters. Our work is orthogonal to [Bru09], since even if such a clustering is given, the problem still remains to estimate the parameters of each component.

1.4.7 Concurrent and (some) subsequent work

In concurrent and independent work to [DKK$^+$16], Lai, Rao, and Vempala [LRV16] also study high-dimensional agnostic learning. In comparison to [DKK$^+$16], their results work for more general types of distributions, but our guarantees are stronger when learning a Gaussian. In particular, their estimates lose factors which are logarithmic in the dimension, whereas our guarantees are always dimension-free. Moreover, their results are superceded than those given in [DKK$^+$17].

After the initial publication of [DKK$^+$16], there has been a flurry of recent work on robust high-dimensional estimation, besides the ones discussed in this thesis. Diakonikolas, Kane, and Stewart [DKS16c] studied the problem of learning the parameters of a graphical model in the presence of noise, when given its graph theoretic structure. Charikar, Steinhardt, and Valiant [CSV17] developed algorithms that can tolerate a fraction of corruptions greater than a half, under the weaker goal of outputting a small list of candidate hypotheses that contains a parameter set close to the true values. studied sparse mean and covariance estimation in the presence of noise obtaining computationally efficient robust algorithms with sample complexity sublinear in the dimension. Diakonikolas, Kane, and Stewart [DKS17] proved statistical query lower bounds providing evidence that the error guarantees of our robust mean and covariance estimation algorithms are best possible, within constant factors, for efficient algorithms.

Diakonikolas, Kane, and Stewart [DKS18a] studied PAC learning of geometric concept classes (including low-degree polynomial threshold functions and intersections of halfspaces) in the same corruption model as ours, obtaining the first dimension-independent error guarantees for these classes. Steinhardt, Charikar, and Valiant [SCV18] focused on deterministic conditions of a dataset which allow robust estimation to be possible. In our initial publication, we gave explicit deterministic conditions in various settings; by focusing directly on this goal, [SCV18] somewhat relaxed some of these assumptions. Meister and Valiant [MV17] studied learning in a crowdsourcing model, where the fraction of honest workers may be very small (similar to [CSV17]). Qiao and Valiant [QV18] considered robust estimation of discrete distributions in a setting where we have several sources (a fraction of which are adversarial) who each provide a batch of samples. Concurrent to [HL18], which we discuss in this thesis, number of simultaneous works [KS18, DKS18b] investigated robust mean estimation in even more general settings, and apply their techniques to learning mixtures of Gaussians under minimal separation conditions. Finally, concurrent to [DKK$^+$18b], a number of results study robustness in supervised learning tasks [PSBR18, KKM18], including regression and SVM problems. Despite all of this rapid progress, there are still many interesting theoretical and practical questions left to explore.

Chapter 2

Convex Programming I: Learning a Gaussian

The pink lights reflecting off of
the waves are so beautiful
Can you hear the trembling sounds
that connect you and I?

In this chapter we present our first framework for robust learning, namely *unknown convex programming.* These algorithms will typically assign weights to individual data points, corresponding to how much the algorithm believes that the data point is good or bad. Then algorithm then hopes to converge to a set of weights which is essentially uniform over the good points. Algorithms based loosely on these sorts of ideas will be the focus on the next three chapters of this thesis. We will show that variants of this general technique can provide polynomial time algorithms for a number of problems in robust learning and beyond.

Naively, given an ε-corrupted set of data points of size n, because we know that at most εn of these data points are bad, a natural set to attempt to optimize over would be the collection of sets of these data points of size $(1-\varepsilon)n$. However, this collection does not inherently possess any convex structure and as a result is difficult to directly optimize over. Insetead, we will have to take some sort of convex relaxation of this set. In the next two chapters, we will use a fairly naive way of relaxing the set of weights, which already turns out to be sufficient for the purposes of these chapters. We will show that the spectral signatures described in the introduction will allow us to efficiently optimize over these relaxed sets. In Chapter 4, we will use a more general relaxation, namely, a relaxation corresponding to the powerful Sum of Squares hierarchy. This will prove vital for solving the problems considered in that chapter.

While these algorithms are polynomial time, in general, the focus of these chapters will be on sample complexity. As we shall see, the correctness of these algorithms does not require very powerful concentration, and these algorithms are often sample-optimal. Morally, it seem that the powerful algorithmic tools we use allow us to argue correctness using subtle, but simple statements (a fact that will prove very crucial in Chapter 4). The framework presented in Chapter 5 and beyond will be significantly more efficient, but will require more delicate concentration bounds to hold. As a result, the analysis for the latter algorithms tend to be more complicated (at least in the author's view), although in the end we are able to get very similar sample complexities in many cases for the two algorithms (up to polylog factors).

2.1 Preliminaries

2.1.1 The Set $S_{n,\varepsilon}$

An important algorithmic object for us will be the following set:

Definition 2.1.1. For any $\frac{1}{2} > \varepsilon > 0$ and any integer n, let

$$S_{n,\varepsilon} = \left\{ (w_1, \ldots, w_n) : \sum_{i=1}^{n} w_i = 1, \text{ and } 0 \leq w_i \leq \frac{1}{(1-\varepsilon)n}, \forall i \right\} .$$

Next, we motivate this definition. For any $J \subseteq [n]$, let $w^J \in \mathbb{R}^n$ be the vector which is given by $w^J_i = \frac{1}{|J|}$ for $i \in J$ and $w^J_i = 0$ otherwise. Then, observe that

$$S_{n,\varepsilon} = \mathrm{conv} \left\{ w^J : |J| = (1-\varepsilon)n) \right\} ,$$

and so we see that this set is designed to capture the notion of selecting a set of $(1-\varepsilon)n$ samples from n samples.

Given $w \in S_{n,\varepsilon}$ we will use the following notation

$$w_g = \sum_{i \in S_{\text{good}}} w_i \text{ and } w_b = \sum_{i \in S_{\text{bad}}} w_i$$

to denote the total weight on good and bad points respectively. The following facts are immediate from $|S_{\text{bad}}| \leq \varepsilon n$ and the properties of $S_{n,\varepsilon}$.

Fact 2.1.1. *If $w \in S_{n,\varepsilon}$ and $|S_{\text{bad}}| \leq \varepsilon n$, then $w_b \leq \frac{\varepsilon}{1-\varepsilon}$. Moreover, the renormalized weights w' on good points given by $w'_i = \frac{w_i}{w_g}$ for all $i \in S_{\text{good}}$, and $w'_i = 0$ otherwise, satisfy $w' \in S_{n,2\varepsilon}$.*

2.1.2 The Ellipsoid algorithm and approximate separation oracles

Throughout this section, our algorithms will build off the ellipsoid algorithm for convex optimization, which we review here. We will first require the notion of a separation oracle for a convex set, which we will slightly generalize later:

Definition 2.1.2. Let $C \subseteq \mathbb{R}^d$ be a convex set. A *separation oracle* for C is an algorithm which, given $x \in \mathbb{R}^d$, either outputs:

- "YES", if $x \in C$, or
- a hyperplane $\ell : \mathbb{R}^d \to \mathbb{R}$ so that $\ell(x) \geq 0$ but $\ell(z) < 0$ for all $z \in C$.

It can be shown that if $x \notin C$, then such a ℓ always exists. It can be shown that such an oracle suffices for (approximately) finding a point in a convex set:

Theorem 2.1.2 ([GLS88]). *Let $R \geq \varepsilon > 0$ be fixed. Let C be a convex set in $\mathbb{R}^d$ so that $C \subseteq B(0, R)$. Suppose there exists a separation oracle $\mathcal{O}$ for C. Then, there exists an algorithm* $\textsc{Ellipsoid}(\mathcal{O}, \varepsilon)$ *which requires* $\mathrm{poly}(d, \log(R/\varepsilon))$ *calls to $\mathcal{O}$, and finds a point x' so that $\|x' - x\|_2 < \varepsilon$ for some $x \in C$.*

In fact, this result can be strengthened to accomodate slightly weaker notions of separation oracle, which will be crucial for us. Specifically, we will require the following notion of an *approximate* separation oracle:

Definition 2.1.3. Let $C \subseteq \mathbb{R}^d$ be a convex set. An *approximate separation oracle* for C is an algorithm which, given $x \in \mathbb{R}^d$, either outputs:

- "YES", if $x \in C'$, or
- a hyperplane $\ell : \mathbb{R}^d \to \mathbb{R}$ so that $\ell(x) \geq 0$ but $\ell(z) < 0$ for all $z \in C'$, if $x \notin C$.

Here $C' \subseteq C$ is some fixed convex set. Moreover, if the algorithm ever outputs a separation oracle, then $\ell(x) < 0$ for all $x \in C'$.

Specifically, the behavior of such an oracle is somewhat unspecified if $x \in C \setminus C'$: it can either output "YES" or a hyperplane. However, any hyperplane output by this algorithm is *always* a separating hyperplane for C'. Then, it can be shown (by the same arguments as in [GLS88]) that this still suffices to approximately find a feasible point in C:

Corollary 2.1.3. *Let $R \geq \varepsilon > 0$ be fixed. Let C be a convex set in $\mathbb{R}^d$ so that $C \subseteq B(0, R)$. Suppose there exists an approximate separation oracle $\mathcal{O}$ for C. Then, there exists an algorithm* $\textsc{Ellipsoid}(\mathcal{O}, \varepsilon)$ *which requires* $\mathrm{poly}(d, \log(R/\varepsilon))$ *calls to $\mathcal{O}$, and finds a point x' so that $\|x' - x\|_2 < \varepsilon$ for some $x \in C$.*

Remark 2.1.1. For the expert, the correctness of the ellipsoid algorithm with this approximate separation oracle follows because outside C, the separation oracle acts exactly as a separation oracle for C'. Thus, as long as the algorithm continues to query points outside of C, the action of the algorithm is equivalent to one with a separation oracle for C'. Moreover, the behavior of the algorithm is such that it will never exclude C', even if queries are made within C. By terminating therefore in $\mathrm{poly}(d, \log(R/\varepsilon))$ steps, from these two conditions, it is clear from the classical theory presented in [GLS88] that the ellipsoid method satisfies the guarantees given above.

For conciseness, throughout this chapter we will often drop the "approximate" and refer to an approximate separation oracle as a separation oracle. Because of the inherent noise in estimation problems, due to variance in the (uncorrupted) samples, all of our separation oracles will be approximate, usually with a single point being the C' in the definition above.

2.1.3 Concentration inequalities

Throughout this section we will make use of various concentration bounds on low moments of Gaussian random variables. Some are well-known, and others are new but follow from known bounds and appropriate union bound arguments.

Empirical estimates of first and second Moments

Here we will give rates of convergence for various statistics for sub-Gaussian distributions with covariance matrix I that we will make use of later. First, we will require the following well-known "per-vector" and "per-matrix" concentration bounds:

Lemma 2.1.4 (Chernoff inequality)**.** *Let n be a positive integer. Let D be a sub-gaussian distribution with mean 0 and covariance I. Let $Y_i \sim D$ be independent, for $i = 1, \ldots, n$. Let $v \in \mathbb{R}^d$ be an arbitrary unit vector. Then, there exist a universal constant $B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n}\langle v, Y_i\rangle\right| > t\right] \le 4\exp\left(-Bnt^2\right) \ .$$

Lemma 2.1.5 (Hanson-Wright)**.** *Let n be a positive integer. Let D be a sub-gaussian distribution with mean 0 and covariance $\Sigma \preceq I$. Let $Y_i \sim D$ be independent, for $i = 1, \ldots, n$. Let $U \in \mathbb{R}^{d\times d}$ satisfy $U \succeq 0$ and $\|U\|_F = 1$. Then, there exists a universal constant $B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n}\mathrm{tr}(X_i X_i^\top U) - \mathrm{tr}(U)\right| > t\right] \le 4\exp\left(-Bn\min(t, t^2)\right) \ .$$

By standard union bound arguments (see e.g. [Ver10]), we obtain the following concentration results for the empirical mean and covariance:

Lemma 2.1.6. *Let n be a positive integer. Let D be a sub-gaussian distribution with mean* 0 *and covariance I. Let $Y_i \sim D$ be independent, for $i = 1, \ldots, n$. Then, there exist universal constants $A, B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^{n} Y_i\right\|_2 > t\right] \leq 4\exp\left(Ad - Bnt^2\right) \; .$$

Lemma 2.1.7. *With the same setup as in Lemma 2.1.6, there exist universal constants $A, B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^{n} Y_i Y_i^\top - I\right\|_2 > t\right] \leq 4\exp\left(Ad - Bn\min(t, t^2)\right) \; .$$

We will also be interested in how well various statistics concentrate around their expectation, when we take the worst-case set of weights in $S_{n,\varepsilon}$. This is more subtle because as we take more samples, any fixed statistic (e.g. taking the uniform distribution over the samples) concentrates better but the size of $S_{n,\varepsilon}$ (e.g. the number of sets of $(1-\varepsilon)n$ samples) grows too.

Lemma 2.1.8. *Let D be a sub-gaussian distribution with mean* 0 *and covariance I. Fix ε and $\delta \leq 1$. There is a $\gamma_1 = O(\varepsilon \log 1/\varepsilon)$ such that if $Y_1, \ldots, Y_n$ are independent samples from D and*

$$n = \Omega\left(\frac{d + \log(1/\delta)}{\gamma_1^2}\right) ,$$

then

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\|\sum_{i=1}^{n} w_i Y_i Y_i^\top - I\right\|_2 \geq \gamma_1\right] \leq \delta \; . \tag{2.1}$$

Before we start the proof, we note that this proof technique will be used a number of times in the next several chapters, and (in the author's humble opinion) is worth understanding, as it provides good insight into the geometry which governs the quantitative guarantees that our algorithms provide.

Proof of Lemma 2.1.8. Recall that for any $J \subseteq [n]$, we let $w^J \in \mathbb{R}^n$ be the vector which is given by $w^J_i = \frac{1}{|J|}$ for $i \in J$ and $w^J_i = 0$ otherwise. By convexity, it suffices to show that

$$\Pr\left[\exists J : |J| = (1-\varepsilon)n, \text{ and } \left\|\sum_{i=1}^n w^J_i Y_i Y_i^\top - (1-\varepsilon)I\right\|_2 \geq \gamma_1\right] \leq \delta \ .$$

For any fixed w^J we have

$$\begin{aligned}\sum_{i=1}^n w^J_i Y_i Y_i^\top - I &= \frac{1}{(1-\varepsilon)n}\sum_{i\in J} Y_i Y_i^\top - I \\ &= \frac{1}{(1-\varepsilon)n}\sum_{i=1}^n Y_i Y_i^\top - \frac{1}{1-2\varepsilon}I \\ &\quad - \left(\frac{1}{(1-\varepsilon)n}\sum_{i\notin J} Y_i Y_i^\top - \left(\frac{1}{1-\varepsilon} - 1\right)I\right) \ .\end{aligned}$$

Therefore, by the triangle inequality, we have

$$\begin{aligned}\left\|\sum_{i=1}^n w^I_i Y_i Y_i^\top - (1-\varepsilon)I\right\|_2 \leq & \left\|\frac{1}{(1-\varepsilon)n}\sum_{i=1}^n Y_i Y_i^\top - \frac{1}{1-\varepsilon}I\right\|_2 \\ & + \left\|\frac{1}{(1-\varepsilon)n}\sum_{i\notin J} Y_i Y_i^\top - \left(\frac{1}{1-\varepsilon} - 1\right)I\right\|_2 \ .\end{aligned}$$

Observe that the first term on the right hand side does not depend on the choice of J. Let E_1 denote the event that

$$\left\|\frac{1}{(1-\varepsilon)n}\sum_{i=1}^n Y_i Y_i^\top - \frac{1}{1-\varepsilon}I\right\|_2 \leq \gamma_1 \ . \tag{2.2}$$

By Lemma 2.1.7, this happens with probability $1-\delta$ so long as

$$n = \Omega\left(\frac{d + \log(1/\delta)}{\gamma_1^2}\right) \ .$$

For any $J \subset [n]$ so that $|J| = (1-\varepsilon)n$, let $E_2(J)$ denote the event that

$$\left\| \frac{1}{(1-\varepsilon)n} \sum_{i \notin J} Y_i Y_i^\top - \left(\frac{1}{1-\varepsilon} - 1 \right) I \right\|_2 \leq \gamma_1 \ .$$

Fix any such J. By multiplying both sides by $\rho = (1-\varepsilon)/\varepsilon$, the event $E_2(J)$ is equivalent to the event that

$$\left\| \frac{1}{\varepsilon n} \sum_{i \notin J} Y_i Y_i^\top - I \right\|_2 > \rho \gamma_1 \ .$$

Let A, B be as in Lemma 2.1.7. Observe that $\rho\gamma_1 = \Omega(\log 1/\varepsilon) \geq 1$ for ε sufficiently small. Then, by Lemma 2.1.7, we have that for any fixed J,

$$\Pr\left[\left\| \frac{1}{\varepsilon n} \sum_{i \notin J} Y_i Y_i^\top - I \right\|_2 > \rho \gamma_1 \right] \leq 4 \exp\left(Ad - B\varepsilon n \rho \gamma \right) \ .$$

Let $H(\varepsilon)$ denote the binary entropy function. We now have

$$\begin{aligned} &\Pr\left[\left(\bigcap_{J : |J| = (1-\varepsilon)n} E_2(J) \right)^c \right] \\ &\overset{(a)}{\leq} 4 \exp\left(\log \binom{n}{\varepsilon n} + Ad - B\varepsilon n \rho \gamma \right) \\ &\overset{(b)}{\leq} 4 \exp\left(nH(\varepsilon) + Ad - B\varepsilon n \rho \gamma \right) \\ &\overset{(c)}{\leq} 4 \exp\left(\varepsilon n (O(\log 1/\varepsilon) - n\rho) + Ad \right) \\ &\overset{(d)}{\leq} 4 \exp\left(-\varepsilon n/2 + Ad \right) \overset{(e)}{\leq} O(\delta) \ , \end{aligned}$$

as claimed, where (a) follows by a union bound over all sets J of size $(1-\varepsilon)n$, (b) follows from the bound $\log \binom{n}{\varepsilon n} \leq \varepsilon H(\varepsilon)$, (c) follows since $H(\varepsilon) = O(\varepsilon \log 1/\varepsilon)$ as $\varepsilon \to 0$, (d) follows from our choice of γ, and (e) follows from our choice of n. This completes the proof. □

A nearly identical argument (Using Chernoff instead of Bernstein in the above proof) yields:

Lemma 2.1.9. *Fix D, ε and δ as above. There is a $\gamma_2 = O(\varepsilon\sqrt{\log 1/\varepsilon})$ such that if $Y_1, \ldots, Y_n$ are independent samples from D and*

$$n = \Omega\left(\frac{d + \log(1/\delta)}{\gamma_2^2}\right) ,$$

then

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^{N} w_i Y_i \right\|_2 \geq \delta_2 \right] \leq \delta . \tag{2.3}$$

It is worth noting that in this case, we get a guarantee of $O(\varepsilon\sqrt{\log 1/\varepsilon})$ rather than $O(\varepsilon \log 1/\varepsilon)$ in Lemma 2.1.8. This is simply because the sub-Gaussian concentration bound (i.e. the Chernoff bound) is stronger than the sub-exponential concentration bound (Bernstein's inequality). Note that by Cauchy-Schwarz, this implies:

Corollary 2.1.10. *Fix $D, \varepsilon, \delta, \gamma_2$ as above. Then, if $Y_1, \ldots, Y_n$ are independent samples from D and*

$$n = \Omega\left(\frac{d + \log(1/\delta)}{\gamma_2^2}\right) ,$$

then

$$\Pr\left[\exists v \in \mathbb{R}^d, \exists w \in S_{n,\varepsilon} : \left\| \left(\sum_{i=1}^{n} w_i Y_i \right) v^\top \right\|_2 \geq \gamma_2 \|v\|_2 \right] \leq \delta . \tag{2.4}$$

We will also require the following, well-known concentration, which says that no sample from a Gaussian deviates too far from its mean in ℓ_2-distance.

Fact 2.1.11. *Let D be a sub-gaussian distribution with mean 0 and covariance I. Fix $\delta > 0$. Let $X_1, \ldots, X_n \sim D$. Then, with probability $1 - \delta$, we have that $\|X_i\|_2 \leq O\left(\sqrt{d \log(n/\delta)}\right)$ for all $i = 1, \ldots, n$.*

Estimation error in the Frobenius norm

Let $X_1, ..., X_n$ be n i.i.d. samples from $\mathcal{N}(0, I)$. In this section we demonstrate a tight bound on how many samples are necessary such that the sample covariance is close to I in Frobenius norm. Let $\widehat{\Sigma}$ denote the empirical second moment matrix, defined to be

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^\top \ .$$

By self-duality of the Frobenius norm, we know that

$$\begin{aligned} \|\widehat{\Sigma} - I\|_F &= \sup_{\|U\|_F=1} \left| \left\langle \widehat{\Sigma} - I, U \right\rangle \right| \\ &= \sup_{\|U\|_F=1} \left| \frac{1}{n} \sum_{i=1}^{n} \mathrm{tr}(X_i X_i^\top U) - \mathrm{tr}(U) \right| \ . \end{aligned}$$

Since there is a 1/4-net over all PSD matrices with Frobenius norm 1 of size 9^{d^2} (see e.g. Lemma 1.18 in [RH17]), the Vershynin-type union bound argument combined with Lemma 2.1.5 immediately gives us the following:

Corollary 2.1.12. *There exist universal constants $A, B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left\| \frac{1}{n} \sum_{i=1}^{n} X_i X_i^\top - I \right\|_F > t \right] \leq 4 \exp\left(A d^2 - B n \min(t, t^2) \right) \ .$$

By the same union bound technique as used in the proof of Lemma 2.1.8, we obtain:

Corollary 2.1.13. *Fix $\varepsilon, \delta > 0$. There is a $\gamma_1 = O(\varepsilon \log 1/\varepsilon)$ such that if $X_1, \ldots, X_n$ are independent samples from $\mathcal{N}(0, I)$, with*

$$n = \Omega\left(\frac{d^2 + \log 1/\delta}{\gamma_1^2} \right) \ ,$$

then

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^{n} w_i X_i X_i^\top - I \right\|_F \geq \gamma_1 \right] \leq \delta \ .$$

Since the proof is essentially identical to the proof of Lemma 2.1.8, we omit the proof. In fact, the proof technique there can be used to show something slightly stronger, which we will require later. The technique actually shows that if we take any set of size at most εn, and take the uniform weights over that set, then the empirical covariance is not too far away from the truth. More formally:

Corollary 2.1.14. *Fix $\varepsilon, \delta > 0$. There is a $\gamma_2 = O(\varepsilon \log 1/\varepsilon)$ such that if $X_1, \ldots, X_n$ are independent samples from $\mathcal{N}(0, I)$, with*

$$n = \Omega\left(\frac{d^2 + \log 1/\delta}{\gamma_2^2}\right) ,$$

then

$$\Pr\left[\exists T \subseteq [n] : |T| \leq \varepsilon n \text{ and } \left\| \sum_{i \in T} \frac{1}{|T|} X_i X_i^\top - I \right\|_F \geq O\left(\gamma_2 \frac{n}{|T|}\right)\right] \leq \delta .$$

We prove this corollary in the Appendix.

Understanding the fourth moment tensor

Our algorithms will be based on understanding the behavior of the fourth moment tensor of a Gaussian when restricted to various subspaces.

The key result in this section is the following:

Theorem 2.1.15. *Let $X \sim \mathcal{N}(0, \Sigma)$. Let M be the $d^2 \times d^2$ matrix given by $M = \mathbb{E}[(X \otimes X)(X \otimes X)^\top]$. Then, as an operator on $\mathcal{S}_{\mathrm{sym}}$, we have*

$$M = 2\Sigma^{\otimes 2} + \left(\Sigma^\flat\right)\left(\Sigma^\flat\right)^\top .$$

It is important to note that the two terms above are *not* the same; the first term is high rank, but the second term is rank one. The proof of this theorem will require Isserlis' theorem, and is deferred to Appendix B.

Concentration of the fourth moment tensor

We also need to show that the fourth moment tensor concentrates:

Theorem 2.1.16. *Fix $\varepsilon, \delta > 0$. There is a $\gamma_3 = O(\varepsilon \log^2 1/\varepsilon)$ so that if $Y_i \sim \mathcal{N}(0, I)$ are independent, for $i = 1, \ldots, n$, where we have*

$$n = \widetilde{\Omega}\left(\frac{d^2 \log^5 1/\delta}{\gamma_3^2}\right) ,$$

and we let $Z_i = Y_i^{\otimes 2}$ and we let $M_4 = \mathbb{E}[Z_i Z_i^\top]$ be the canonical flattening of the true fourth moment tensor, then we have

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^n w_i Z_i Z_i^\top - M_4 \right\|_{\mathcal{S}} \geq \gamma_3 \right] \leq \delta .$$

To do so will require somewhat more sophisticated techniques than the ones used so far to bound spectral deviations. At a high level, this is because fourth moments of Gaussians have a sufficiently larger variance that the union bound techniques used so far are insufficient. However, we will show that the tails of degree four polynomials of Gaussians still sufficiently concentrate such that removing points cannot change the mean by too much. The proof requires slightly fancy machinery and appears in Appendix E.

2.2 Learning a Gaussian robustly via convex programming

This section is dedicated to one of two efficient algorithms for solving Problem 1.4.1 and its two sub-problems, Problem 1.4.2 and Problem 1.4.3. Specifically, our results are the following:

Theorem 2.2.1. *Fix $\varepsilon, \delta > 0$, and let $\mu \in \mathbb{R}^d$ and let $\Sigma \in \mathbb{R}^{d \times d}$ be positive definite. Given an ε-corrupted set of samples of size n from $\mathcal{N}(\mu, \Sigma)$, where*

$$n = \widetilde{\Omega}\left(\frac{d^2 \log^5(1/\delta)}{\varepsilon^2}\right) ,$$

there is an efficient algorithm which outputs $\widehat{\mu}, \widehat{\Sigma}$ so that with probability $1 - \delta$ we *have*

$$d_{\mathrm{TV}}\left(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\widehat{\mu},\widehat{\Sigma})\right)\leq O(\varepsilon\log 1/\varepsilon)\ .$$

To do this, we solve the robust mean and covariance estimation problem separately. For each of these subproblems we achieve the following guarantees. For mean estimation, we achieve:

Theorem 2.2.2. *Fix $\varepsilon,\delta>0$, and let $\mu\in\mathbb{R}^d$. Given an ε-corrupted set of samples of size n from $\mathcal{N}(\mu,I)$, where*

$$n=\Omega\left(\frac{d+\log(1/\delta)}{\varepsilon^2\log 1/\varepsilon}\right)\ ,$$

there is an efficient algorithm which outputs $\widehat{\mu}$ so that with probability $1-\delta$ we have $\|\mu-\widehat{\mu}\|_2<O(\varepsilon\sqrt{\log 1/\varepsilon})$.

We remark that this result can be easily generalized to general isotropic sub-Gaussian distributions:

Theorem 2.2.3. *Fix $\varepsilon,\delta>0$, and let $\mu\in\mathbb{R}^d$. Given an ε-corrupted set of samples of size n from D, where D is a sub-Gaussian distribution with covariance matrix I, where*

$$n=\Omega\left(\frac{d+\log(1/\delta)}{\varepsilon^2\log 1/\varepsilon}\right)\ ,$$

there is an efficient algorithm which outputs $\widehat{\mu}$ so that with probability $1-\delta$ we have $\|\mu-\widehat{\mu}\|_2<O(\varepsilon\sqrt{\log 1/\varepsilon})$.

For covariance estimation, we achieve:

Theorem 2.2.4. *Fix $\varepsilon,\delta>0$, and let $\Sigma\in\mathbb{R}^{d\times d}$ be positive definite. Given an ε-corrupted set of samples of size n from $\mathcal{N}(0,\Sigma)$, where*

$$n=\widetilde{\Omega}\left(\frac{d^2\log^5(1/\delta)}{\varepsilon^2}\right)\ ,$$

there is an efficient algorithm which outputs $\widehat{\Sigma}$ *so that with probability* $1-\delta$ *we have* $\|\Sigma - \widehat{\Sigma}\|_\Sigma < O(\varepsilon \log 1/\varepsilon)$.

We pause here to make a couple of remarks. First, we note that the mean estimation algorithm easily generalizes to learn the mean of sub-Gaussian distributions with identity covariance. Generalizing the results to sub-Gaussian distributions where we only have an upper bound on the covariance is more difficult. In Chapter 4 we make partial progress on this problem.

To the best of our knowledge, the covariance estimation algorithms do not easily generalize to many other settings. This is because the covariance estimation algorithm heavily leverages the algebraic structure that higher moments of Gaussians have.

We also remark that for both of these settings, the sample complexity we obtain for the robust versions of the problem matches the sample complexity of non-agnostic learning, up to logarithmic factors. That is, it is a folklore result that even without noise, given sample access to $\mathcal{N}(\mu, I)$, to obtain an estimator $\widehat{\mu}$ which satisfies $\mathbb{E}[\|\mu - \widehat{\mu}\|_2] \leq \varepsilon$ requires $n = \Omega(d/\varepsilon^2)$ samples. Similarly, given sample access to $\mathcal{N}(0, \Sigma)$, to obtain an estimator $\widehat{\Sigma}$ which satisfies $\mathbb{E}[\|\widehat{\Sigma} - \Sigma\|_\Sigma] \leq \varepsilon$ requires $n = \Omega(d^2/\varepsilon^2)$ samples. In fact, for mean estimation of an isotropic sub-Gaussian random variable, we are able to exactly match the rate achievable in the non-robust setting, up to constants.

While often in robust statistics, sample complexity is considered a secondary concern[1], we note that the type of concentration that yields these sorts of rates will prove to be very important in our analysis. This is because, intuitively, these concentration inequalities imply that the empirical statistics still converge even when an ε-fraction of the points are removed. This is what allows us to prove Lemma 2.1.8, for instance, which is crucial for our algorithm.

2.2.1 Finding the mean, using a separation oracle

In this section, we consider the problem of approximating μ given an ε-corrupted set of n samples from $\mathcal{N}(\mu, I)$. We remark that everything here generalizes trivially to the

[1] Orthogonally, the author believes that this lack of concern regarding sample complexity is unfortunate; such rates often govern how useful the methods will be in practice!

setting where the distribution is a sub-Gaussian distribution with identity covariance, so for simplicity of exposition, we will only consider the case where the distribution is Gaussian. Throughout this section, we will let $\mu \in \mathbb{R}^d$ be the true (unknown) mean, and we let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$.

Our algorithm will be based on working with the following convex set:

$$\mathcal{C}_\gamma = \left\{ w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^n w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 \leq \gamma \right\} .$$

It is not hard to show that $\mathcal{C}_\gamma$ is non-empty for reasonable values of γ (and we will show this later). Moreover we will show that for any set of weights w in $\mathcal{C}_\gamma$, the empirical average

$$\widehat{\mu} = \widehat{\mu}(w) = \sum_{i=1}^n w_i X_i$$

will be a good estimate for μ. The challenge is that since μ itself is unknown, there is not an obvious way to design a separation oracle for $\mathcal{C}_\gamma$ even though it is convex. Our algorithm will run in two basic steps. First, it will run a very naive outlier detection to remove any points which are more than $O(\sqrt{d})$ away from the good points. These points are sufficiently far away that a very basic test can detect them. Then, with the remaining points, it will use the approximate separation oracle given below to approximately optimize with respect to C_γ. It will then take the outputted set of weights and output the empirical mean with these weights. We will explain these steps in detail below.

Deterministic Conditions We first lay out a set of determinstic conditions under which our algorithm will work. Specifically, we will require:

$$\|X_i - \mu\|_2 \leq O\left(\sqrt{d \log(n/\delta)}\right), \forall i \in S_{\mathrm{good}} \;, \tag{2.5}$$

$$\left\| \sum_{i \in S_{\mathrm{good}}} w_i (X_i - \mu)(X_i - \mu)^\top - w_g I \right\|_2 \leq \gamma_1 \; \forall w \in S_{n,2\varepsilon}, \text{ and} \tag{2.6}$$

$$\left\| \sum_{i \in S_{\mathrm{good}}} w_i (X_i - \mu) \right\|_2 \leq \gamma_2 \; \forall w \in S_{n,2\varepsilon} \;\;, \tag{2.7}$$

where

$$\gamma_1 = O(\varepsilon \log 1/\varepsilon), \text{ and } \gamma_2 = O(\varepsilon \sqrt{\log 1/\varepsilon}) \;.$$

The concentration bounds we gave earlier were exactly bounds on the failure probability of either of these conditions, albeit for $S_{n,\varepsilon}$ instead of $S_{n,2\varepsilon}$. Thus, by increasing ε by a constant factor we get the same sorts of concentration guarantees. Formally, we have:

Corollary 2.2.5. *Fix $\varepsilon, \delta > 0$, and let $\gamma = O(\varepsilon\sqrt{\log 1/\varepsilon})$. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$, where*

$$n = \Omega\left(\frac{d + \log 1/\delta}{\gamma^2}\right) \;.$$

Then, (2.5)*-*(2.7) *hold simultaneously with probability at least $1-\delta$, with $\gamma_1 = O(\varepsilon \log 1/\varepsilon)$ and $\gamma_2 = O(\varepsilon\sqrt{\log 1/\varepsilon})$.*

Proof. This follows by Fact 2.1.11, Lemma 2.1.8, Lemma 2.1.9 and a union bound. □

Naive pruning

The first step of our algorithm will be to remove points which have distance which is much larger than $O(\sqrt{d})$ from the mean. Our algorithm is very naive: it computes all pairwise distances between points, and throws away all points which have distance more than $O(\sqrt{d})$ from more than a 2ε-fraction of the remaining points.

Algorithm 1 Naive Pruning

```
function NaivePrune($X_1, \ldots, X_n$)
    For $i, j = 1, \ldots, n$, define $\gamma_{i,j} = \|X_i - X_j\|_2$.
    for $i = 1, \ldots, j$ do
        Let $A_i = \{j \in [n] : \gamma_{i,j} > \Omega(\sqrt{d \log(n/\delta)})\}$
        if $|A_i| > 2\varepsilon n$ then
            Remove $X_i$ from the set.
    return the pruned set of samples.
```

Then we have the following fact:

Fact 2.2.6. *Suppose that (2.5) holds. Then* NaivePrune *removes no uncorrupted points, and moreover, if X_i is not removed by* NaivePrune*, we have $\|X_i - \mu\|_2 \leq O\left(\sqrt{d \log(n/\delta)}\right)$.*

Proof. That no uncorrupted point is removed follows directly from (2.5) and the fact that there can be at most $2\varepsilon n$ corrupted points. Similarly, if X_i is not removed by NaivePrune, that means there must be an uncorrupted X_j such that $\|X_i - X_j\|_2 \leq O(\sqrt{d \log(n/\delta)})$. Then the desired property follows from (2.5) and a triangle inequality. □

Henceforth, for simplicity we shall assume that no point was removed by NaivePrune, and that for all $i = 1, \ldots, n$, we have $\|X_i - \mu\|_2 < O(\sqrt{d \log(n/\delta)})$. Otherwise, we can simply work with the pruned set, and it is evident that nothing changes.

The separation oracle

Our main result in this section is an approximate separation oracle for $\mathcal{C}_\gamma$. Observe that technically, for the ellipsoid algorithm, we need a separation oracle for arbitrary w, not just $w \in S_{n,\varepsilon}$. However, since it is trivial to construct a separation oracle for $S_{n,\varepsilon}$, we will only focus on the case where $w \in S_{n,\varepsilon}$. Thus, throughout this section, let $w \in S_{n,\varepsilon}$ and set $\widehat{\mu} = \widehat{\mu}(w) = \sum_{i=1}^n w_i X_i$. Let $\Delta = \mu - \widehat{\mu}$. Our first step is to show the following key lemma, which states that any set of weights that does not yield a good estimate for μ cannot be in the set $\mathcal{C}_\gamma$:

Lemma 2.2.7. *Suppose that (2.6)-(2.7) holds. Suppose that* $\|\Delta\|_2 \geq \Omega(\gamma_2)$. *Then*

$$\left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 \geq \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon} \right) .$$

We pause to remark that this lemma is a very concrete formalization of the notion of spectral signatures mentioned in the introduction. It says that if the empirical mean has been corrupted, then the spectral norm of the empirical covariance must be large. This immediately gives us a way to check if the empirical mean has been corrupted (namely, by checking the empirical covariance). In a certain sense, the rest of this section will be devoted to converting this detection guarantee into a optimization routine.

Proof. By Fact 2.1.1 and (2.7) we have $\| \sum_{i \in S_{\text{good}}} \frac{w_i}{w_g} X_i - \mu \|_2 \leq \gamma_2$. Now by the triangle inequality we have

$$\left\| \sum_{i \in S_{\text{bad}}} w_i (X_i - \mu) \right\|_2 \geq \|\Delta\|_2 - \left\| \sum_{i \in S_{\text{good}}} w_i (X_i - \mu) - w_g \mu \right\|_2 \geq \Omega(\|\Delta\|_2)$$

Using the fact that the variance is nonnegative we have

$$\sum_{i \in S_{\text{bad}}} \frac{w_i}{w_b} (X_i - \mu)(X_i - \mu)^\top \succeq \left(\sum_{i \in S_{\text{bad}}} \frac{w_i}{w_b} (X_i - \mu) \right) \left(\sum_{i \in S_{\text{bad}}} \frac{w_i}{w_b} (X_i - \mu) \right)^\top ,$$

and therefore

$$\left\| \sum_{i \in S_{\text{bad}}} w_i (X_i - \mu)(X_i - \mu)^\top \right\|_2 \geq \Omega\left(\frac{\|\Delta\|_2^2}{w_b} \right) \geq \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon} \right) .$$

On the other hand,

$$\begin{aligned} \left\| \sum_{i \in S_{\text{good}}} w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 &\leq \left\| \sum_{i \in S_{\text{good}}} w_i (X_i - \mu)(X_i - \mu)^\top - w_g I \right\|_2 + w_b \\ &\leq \gamma_1 + w_b . \end{aligned}$$

where in the last inequality we have used Fact 2.1.1 and (2.6). Hence altogether this implies that

$$\left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 \geq \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right) - w_b - \gamma_1 \geq \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right) ,$$

since $\Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right) = \Omega(\varepsilon \log 1/\varepsilon) > \gamma_1$. This completes the proof. □

As a corollary, we find that *any* set of weights in $\mathcal{C}_\gamma$ immediately yields a good estimate for μ:

Corollary 2.2.8. *Suppose that (2.6) and (2.7) hold. Let $w \in \mathcal{C}_\gamma$ for $\gamma = O(\varepsilon \log 1/\varepsilon)$. Then*

$$\|\Delta\|_2 \leq O(\varepsilon\sqrt{\log 1/\varepsilon})$$

We now have the tools to give an approximate separation oracle for $\mathcal{C}_\gamma$ with $\gamma = O(\varepsilon \log 1/\varepsilon)$.

Theorem 2.2.9. *Fix $\varepsilon > 0$, and let $\gamma = O(\varepsilon \log 1/\varepsilon)$. Suppose that (2.6) and (2.7) hold. Let w^* denote the weights which are uniform on the uncorrupted points. Then there is a constant $c > 0$ and an algorithm such that:*

1. *(Completeness) If $w = w^*$, then it outputs "YES".*

2. *(Soundness) If $w \notin \mathcal{C}_{c\gamma}$, the algorithm outputs a hyperplane $\ell : \mathbb{R}^N \to \mathbb{R}$ such that $\ell(w) \geq 0$ but $\ell(w^*) < 0$. Moreover, if the algorithm ever outputs a hyperplane ℓ, then $\ell(w^*) < 0$.*

We remark that by Corollary 2.1.3, these two facts imply that for any $\delta > 0$, the ellipsoid method with this separation oracle will output a w' such that $\|w - w'\|_\infty < \varepsilon/(n\sqrt{d \log(n/\delta)})$, for some $w \in \mathcal{C}_{c\gamma}$ in $\mathrm{poly}(d, 1/\varepsilon, \log 1/\delta)$ steps.

The separation oracle is given in Algorithm 2. Next, we prove correctness for our approximate separation oracle:

Algorithm 2 Separation oracle sub-procedure for agnostically learning the mean.

```
function SeparationOracleUnknownMean$(w, \varepsilon, X_1, \dots, X_N)$
    Let $\widehat{\mu} = \sum_{i=1}^n w_i X_i$.
    Let $\gamma = O(\varepsilon \log 1/\varepsilon)$.
    For $i = 1, \dots, n$, define $Y_i = X_i - \widehat{\mu}$.
    Let $\lambda$ be the eigenvalue of largest magnitude of $M = \sum_{i=1}^n w_i Y_i Y_i^\top - I$.
    Let $v$ be its associated eigenvector.
    if $|\lambda| \le \frac{c}{2}\gamma$ then
        return “YES".
    else if $\lambda > \frac{c}{2}\gamma$ then
        return the hyperplane $\ell(u) = (\sum_{i=1}^n u_i \langle Y_i, v \rangle^2 - 1) - \lambda$.
    else
        return the hyperplane $\ell(u) = \lambda - (\sum_{i=1}^n u_i \langle Y_i, v \rangle^2 - 1)$.
```

Proof of Theorem 2.2.9. Again, let $\Delta = \mu - \widehat{\mu}$, and let $M = \sum_{i=1}^N w_i Y_i Y_i^\top - I$. By expanding out the formula for M, we get:

$$\begin{aligned}
\sum_{i=1}^N w_i Y_i Y_i^\top - I &= \sum_{i=1}^N w_i (X_i - \mu + \Delta)(X_i - \mu + \Delta)^\top - I \\
&= \sum_{i=1}^N w_i (X_i - \mu)(X_i - \mu)^\top - I + \sum_{i=1}^N w_i (X_i - \mu)\Delta^\top \\
&\quad + \Delta \sum_{i=1}^N w_i (X_i - \mu)^\top + \Delta\Delta^\top \\
&= \sum_{i=1}^N w_i (X_i - \mu)(X_i - \mu)^\top - I - \Delta\Delta^\top \ .
\end{aligned}$$

Let us now prove completeness.

Claim 2.2.10. *Suppose $w = w^*$. Then $\|M\|_2 < \frac{c}{2}\gamma$.*

Proof. Recall that w^* are the weights that are uniform on the uncorrupted points. Because $|S_{\text{bad}}| \le \varepsilon n$ we have that $w^* \in S_{n,\varepsilon}$. We can now use (2.6) to conclude that $w^* \in \mathcal{C}_{\gamma_1}$. Now by Corollary 2.2.8 we have that $\|\Delta\|_2 \le O(\varepsilon\sqrt{\log 1/\varepsilon})$. Thus

$$\begin{aligned}
\left\| \sum_{i=1}^n w_i^* (X_i - \mu)(X_i - \mu)^\top - I - \Delta\Delta^\top \right\|_2 &\le \left\| \sum_{i=1}^n w_i^* (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 + \|\Delta\Delta^\top\|_2 \\
&\le \gamma_1 + O(\varepsilon^2 \log 1/\varepsilon) < \frac{c\gamma}{2} \ .
\end{aligned}$$

□

We now turn our attention to soundness.

Claim 2.2.11. *Suppose that $w \notin C_{c\gamma}$. Then $|\lambda| > \frac{c}{2}\gamma$.*

Proof. By the triangle inequality, we have

$$\left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I - \Delta\Delta^\top \right\|_2 \geq \left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_2 - \left\| \Delta\Delta^\top \right\|_2 .$$

Let us now split into two cases. If $\|\Delta\|_2 \leq \sqrt{c\gamma/10}$, then the first term above is at least $c\gamma$ by definition and we can conclude that $|\lambda| > c\gamma/2$. On the other hand, if $\|\Delta\|_2 \geq \sqrt{c\gamma/10}$, by Lemma 2.2.7, we have that

$$\left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I - \Delta\Delta^\top \right\|_2 \geq \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right) - \|\Delta\|_2^2 = \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right) . \tag{2.8}$$

which for sufficiently small ε also yields $|\lambda| > c\gamma/2$. □

Now by construction $\ell(w) \geq 0$ (in fact $\ell(w) = 0$). All that remains is to show that $\ell(w^*) < 0$ always holds. We will only consider the case where the top eigenvalue λ of M is positive. The other case (when $\lambda < -\frac{c}{2}\gamma$) is symmetric. We will split the analysis into two parts. We have

$$\begin{aligned} \left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top - I \right\|_2 &= \left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} (X_i - \mu + \Delta)(X_i - \mu + \Delta)^\top - I \right\|_2 \\ &\leq \underbrace{\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} (X_i - \mu)(X_i - \mu)^\top - I \right\|_2}_{\leq \gamma_1} + \underbrace{2\|\Delta\|_2 \left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} (X_i - \mu) \right\|_2}_{\leq 2\gamma_2 \|\Delta\|_2 \text{ by (2.7)}} + \|\Delta\|_2^2 \end{aligned} \tag{2.9}$$

Suppose $\|\Delta\|_2 \leq \sqrt{c\gamma/10}$. By (2.9) we immediately have:

$$\ell(w^*) \leq \gamma_1 + 2\gamma_2\|\Delta\|_2 + \|\Delta\|_2^2 - \lambda \leq \frac{c\gamma}{5} - \lambda < 0 \; ,$$

since $\lambda > c\gamma/2$. On the other hand, if $\|\Delta\|_2 \geq \sqrt{c\gamma/10}$ then by (2.8) we have $\lambda = \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right)$. Putting it all together we have:

$$\ell(w^*) \leq \underbrace{\left\| \frac{1}{|S_{\mathrm{good}}|} \sum_{i \in S_{\mathrm{good}}} (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top - I \right\|_2}_{\leq \gamma_1 + 2\gamma_2\|\Delta\|_2 + \|\Delta\|_2^2} - \lambda \; ,$$

where in the last line we used the fact that $\lambda > \Omega\left(\frac{\|\Delta\|_2^2}{\varepsilon}\right)$, and $\|\Delta\|_2^2 \geq \Omega(\varepsilon^2 \log 1/\varepsilon)$. This now completes the proof. □

The full algorithm

This separation oracle, along Corollary 2.1.3, implies that we have shown the following:

Corollary 2.2.12. *Fix $\varepsilon, \delta > 0$, and let $\gamma = O(\varepsilon\sqrt{\log 1/\varepsilon})$. Let $X_1, \ldots, X_n$ be a set of points satisfying (2.6)-(2.7), for $\gamma_1, \gamma_2 \leq \gamma$. Let c be a sufficiently large constant. Then, there is an algorithm* LearnApproxMean$(\varepsilon, \delta, X_1, \ldots, X_n)$ *which runs in time* $\mathrm{poly}(n, d, 1/\varepsilon, \log 1/\delta)$*, and outputs a set of weights $w' \in S_{n,\varepsilon}$ such that there is a $w \in C_{c\gamma}$ such that $\|w - w'\|_\infty \leq \varepsilon/(n\sqrt{d \log(n/\delta)})$.*

This algorithm, while an extremely powerful primitive, is technically not sufficient. However, given this, the full algorithm is not too difficult to state: simply run NaivePrune, then optimize over $C_{c\gamma}$ using this separation oracle, and get some w which is approximately in $C_{c\gamma}$. Then, output $\sum_{i=1}^n w_i X_i$. For completeness, the pseudocode for the algorithm is given below. In the pseudocode, we assume that Ellipsoid(SeparationOracleUnknownMean, ε') is a convex optimization routine, which given the SeparationOracleUnknownMean separation oracle and a target error ε', outputs a w' such that $\|w - w'\|_\infty \leq \varepsilon'$. From the classical theory of optimization, we know such a routine exists and runs in polynomial time.

Algorithm 3 Convex programming algorithm for agnostically learning the mean.

```
function LearnMean($\varepsilon, \delta, X_1, \ldots, X_n$)
    Run NaivePrune($X_1, \ldots, X_n$). Let $\{X_i\}_{i \in I}$ be the pruned set of samples.
      /* For simplicity assume $I = [n]$ */
    Let $w' \leftarrow$ LearnApproxMean($\varepsilon, \delta, X_1, \ldots, X_N$).
    return $\sum_{i=1}^n w'_i X_i$.
```

We have:

Theorem 2.2.13. *Fix $\varepsilon, \delta > 0$, and let $\gamma = O(\varepsilon\sqrt{\log 1/\varepsilon})$. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$, where*

$$n = \Omega\left(\frac{d + \log 1/\delta}{\gamma^2}\right) .$$

Let $\widehat{\mu}$ be the output of LearnMean$(\varepsilon, \delta, X_1, \ldots, X_n)$. *Then with probability $1 - \delta$, we have $\|\widehat{\mu} - \mu\|_2 \le \gamma$.*

Proof. Condition on the event that (2.5)-(2.7) hold for the original uncorrupted set of points. By Corollary 2.2.5, this happens with probability $1 - \delta$. After NaivePrune, by Fact 2.2.6 we may assume that no uncorrupted points are removed, and all points satisfy $\|X_i - \mu\|_2 \le O(\sqrt{d \log(n/\delta)})$. Let w' be the output of the algorithm, and let $w \in C_{c\gamma}$ be such that $\|w - w'\|_\infty < \varepsilon/(n\sqrt{d\log(n/\delta)})$. By Corollary 2.2.8, we know that $\| \sum_{i=1}^n w_i X_i - \mu\|_2 \le O(\gamma)$. Hence, we have

$$\left\| \sum_{i=1}^n w'_i X_i - \mu \right\|_2 \le \left\| \sum_{i=1}^n w_i X_i - \mu \right\|_2 + \sum_{i=1}^n |w_i - w'_i| \cdot \|X_i - \mu\|_2 \le O(\gamma) + \varepsilon ,$$

so the entire error is at most $O(\gamma)$, as claimed. □

2.2.2 An extension, with small spectral noise

For learning of arbitrary Gaussians, we will need a simple extension that allows us to learn the mean even in the presence of some spectral norm error in the covariance matrix. Since the algorithms and proofs are almost identical to the techniques above, we omit them for conciseness. Formally, we require:

Theorem 2.2.14. *Fix $\chi, \varepsilon, \delta > 0$, let γ be as in Theorem 2.2.13, and let $X_1, \ldots, X_n$ be an ε-corrupted set of points from $\mathcal{N}(\mu, \Sigma)$, where $\|\Sigma - I\|_2 \leq O(\chi)$, and where*

$$n = \Omega\left(\frac{d + \log 1/\delta}{\gamma^2}\right) .$$

There is an algorithm $\textsc{RecoverMeanNoisy}(X_1, \ldots, X_n, \varepsilon, \delta, \gamma, \chi)$ *which runs in time* $\mathrm{poly}(d, 1/\chi, 1/\varepsilon, \log 1/\delta)$ *and outputs a $\widehat{\mu}$ so that with probability $1 - \delta$, we have $\|\widehat{\mu} - \mu\|_2 \leq \gamma + O(\chi)$.*

This extension follows from the observation that we only need spectral guarantees on our covariance matrix, and whatever error we have in these concentration goes directly into our error guarantee. Thus, by the same calculations that we had above, if the eigenvalues are at most $1 + \alpha$, this directly goes linearly into our final error bound.

2.2.3 Finding the covariance, using a separation oracle

In this section, we consider the problem of learning the covariance of a Gaussian given corrupted samples. Throughout this section, we let $\Sigma \in \mathbb{R}^{d \times d}$ be an (unknown) positive definite matrix, and we let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$. Let $U_i = \Sigma^{-1/2} X_i$ such that if $X_i \sim \mathcal{N}(0, \Sigma)$ then $U_i \sim \mathcal{N}(0, I)$. Moreover let $Z_i = U_i^{\otimes 2}$. Our approach will parallel the one given earlier in Section 2.2.1. Again, we will work with a convex set

$$C_\gamma = \left\{ w \in S_{n,\varepsilon} : \left\| \left(\sum_{i=1}^n w_i X_i X_i^\top \right) - \Sigma \right\|_\Sigma \leq \gamma \right\} .$$

and our goal is to design an approximate separation oracle. Our results in this section will rely on the following deterministic conditions:

$$\|U_i\|_2^2 \leq O\left(d \log(n/\delta)\right) \ , \forall i \in S_{\text{good}} \tag{2.10}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i U_i U_i^\top - w_g I \right\|_F \leq \gamma_1 \ , \tag{2.11}$$

$$\left\| \sum_{i \in T} \frac{1}{|T|} U_i U_i^\top - I \right\|_F \leq O\left(\gamma_2 \frac{n}{|T|}\right) \ , \text{and} \tag{2.12}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i Z_i Z_i^\top - w_g M_4 \right\|_{\mathcal{S}} \leq \gamma_3 \ , \tag{2.13}$$

for all $w \in S_{n,\varepsilon}$, and all sets $T \subseteq S_{\text{good}}$ of size $|T| \leq \varepsilon n$. As before, by Fact 2.1.1, the renormalized weights over the uncorrupted points are in $S_{n,2\varepsilon}$. Hence, we can appeal to Fact 2.1.11, Corollary 2.1.13, Corollary 2.1.14, and Theorem 2.1.16 with $S_{n,2\varepsilon}$ instead of $S_{n,\varepsilon}$and get that if we set $\gamma_1, \gamma_2 = O(\varepsilon\sqrt{\log 1/\varepsilon})$ and $\gamma_3 = O(\varepsilon \log^2 1/\varepsilon)$, these conditions simultaneously hold with probability $1 - \delta$. Let w^* be the set of weights which are uniform over the uncorrupted points; by (2.11) for $\gamma \geq \Omega(\varepsilon\sqrt{\log 1/\varepsilon})$ we have that $w^* \in C_\gamma$.

Our main result is that under these conditions, there is an approximate separation oracle for C_γ. As was for mean estimation, in the design of the separation oracle, we will only consider $w \in S_{n,\varepsilon}$, since otherwise the separation oracle is trivial. Formally, we show:

Theorem 2.2.15. *Let $\gamma = O(\varepsilon \log 1/\varepsilon)$. Suppose that (2.11), (2.12), and 2.13 hold for $\gamma_1, \gamma_2 \leq O(\gamma)$ and $\gamma_3 \leq O(\gamma \log 1/\varepsilon)$. Then, there is a constant $c > 0$ and an algorithm such that, given any input $w \in S_{n,\varepsilon}$ we have:*

1. *(Completeness) If $w = w^*$, the algorithm outputs "YES".*

2. *(Soundness) If $w \notin C_{c\gamma}$, the algorithm outputs a hyperplane $\ell : \mathbb{R}^m \to \mathbb{R}$ such that $\ell(w) \geq 0$ but we have $\ell(w^*) < 0$. Moreover, if the algorithm ever outputs a hyperplane ℓ, then $\ell(w^*) < 0$.*

As in the case of learning an unknown mean, by the classical theory of convex optimization this implies that we will find a point w such that $\|w - w'\|_\infty \leq \frac{\varepsilon}{\mathrm{poly}(n)}$ for some $w' \in C_{c\gamma}$, using polynomially many calls to this oracle. We make this more precise in the following subsubsection.

The pseudocode for the (approximate) separation oracle is given in Algorithm 4. Observe briefly that this algorithm does indeed run in polynomial time. Lines 2-7 require only taking top eigenvalues and eigenvectors, and so can be done in polynomial time. For any $\xi \in \{-1, +1\}$, line 8 can be run by sorting the samples by $w_i \left(\frac{\|Y_i\|^2}{\sqrt{d}} - \sqrt{d} \right)$ and seeing if there is a subset of the top $2\varepsilon n$ samples satisfying the desired condition, and line 9 can be executed similarly.

Algorithm 4 Convex programming algorithm for agnostically learning the covariance.

```
 1: function SeparationOracleUnknownCovariance($w$)
 2:     Let $\widehat{\Sigma} = \sum_{i=1}^n w_i X_i X_i^\top$.
 3:     For $i = 1, \ldots, n$, let $Y_i = \widehat{\Sigma}^{-1/2} X_i$ and let $Z_i = (Y_i)^{\otimes 2}$.
 4:     Let $v$ be the top eigenvector of $M = \sum_{i=1}^n w_i Z_i Z_i^\top - 2I$ restricted to $\mathcal{S}$, and
        let $\lambda$ be its associated eigenvalue.
 5:     if $|\lambda| > \Omega(\varepsilon \log^2 1/\varepsilon)$ then
 6:         Let $\xi = \mathrm{sgn}(\lambda)$.
 7:         return the hyperplane
                $$\ell(u) = \xi \left( \sum_{i=1}^n u_i \langle v, Z_i \rangle^2 - 2 - \lambda \right) .$$
 8:     else if there exists a sign $\xi \in \{-1, 1\}$ and a set $T$ of samples of size at most
        $\varepsilon n$ such that
                $$\alpha = \xi \sum_{i \in T} w_i \left( \frac{\|Y_i\|_2^2}{\sqrt{d}} - \sqrt{d} \right) > \frac{(1 - \varepsilon)\alpha\gamma}{2} ,$$
        then
 9:         return the hyperplane
                $$\ell(u) = \xi \sum_{i \in T} u_i \left( \frac{\|Y_i\|_2^2}{\sqrt{d}} - \sqrt{d} \right) - \alpha ,$$
10:     else
11:         return "YES".
```

We now turn our attention to proving the correctness of this separation oracle. We require the following technical lemmata.

Claim 2.2.16. *Let $w_1, \ldots, w_n$ be a set of non-negative weights such that $\sum_{i=1}^n w_i = 1$, and let $a_i \in \mathbb{R}$ be arbitrary. Then*

$$\sum_{i=1}^n a_i^2 w_i \geq \left(\sum_{i=1}^n a_i w_i \right)^2 .$$

Proof. Let P be the distribution where a_i is chosen with probability w_i. Then $\mathbb{E}_{X \sim P}[X] = \sum_{i=1}^n f a_i w_i$ and $\mathbb{E}_{X \sim P}[X^2] = \sum_{i=1}^n a_i w_i^2$. Since $\mathrm{Var}_{X \sim P}[X] = \mathbb{E}_{X \sim P}[X^2] - \mathbb{E}_{X \sim P}[X]^2$ is always a non-negative quantity, by rearranging the desired conclusion follows. □

Lemma 2.2.17. *Fix $\gamma < 1$ and suppose that M is symmetric. If $\|M - I\|_F \geq \gamma$ then $\|M^{-1} - I\|_F \geq \frac{\gamma}{2}$.*

Proof. We will prove this lemma in the contrapositive, by showing that if $\|M^{-1} - I\|_F < \frac{\gamma}{2}$ then $\|M - I\|_F < \gamma$. Since the Frobenius norm is rotationally invariant, we may assume that $M^{-1} = \mathrm{diag}(1 + \nu_1, \ldots, 1 + \nu_d)$, where by assumption $\sum \nu_i^2 < \gamma^2/4$. By our assumption that $\gamma < 1$, we have $|\nu_i| \leq 1/2$ for all i. Thus

$$\sum_{i=1}^d \left(1 - \frac{1}{1 + \nu_i} \right)^2 \leq \sum_{i=1}^d 4\nu_i^2 < \gamma ,$$

where we have used the inequality $|1 - \frac{1}{1+x}| \leq |2x|$ which holds for all $|x| \leq 1/2$. This completes the proof. □

Lemma 2.2.18. *Let $M, N \in \mathbb{R}^{d \times d}$ be arbitrary matrices. Then $\|MN\|_F \leq \|M\|_2 \|N\|_F$.*

Proof. Let $N_1, \ldots, N_d$ be the columns of N. Then

$$\|MN\|_F^2 = \sum_{i=1}^d \|MN\|_2^2 \leq \|M\|_2^2 \sum_{i=1}^d \|N_i\|_2^2 = \|M\|_2^2 \|N\|_F^2 ,$$

so the desired result follows by taking square roots of both sides. □

Lemma 2.2.19. *Let $M \in \mathbb{R}^{d\times d}$. Then,* $\left\| \left(M^\flat\right)\left(M^\flat\right)^\top \right\|_{\mathcal{S}} \leq \|M - I\|_F^2$.

Proof. By the definition of $\|\cdot\|_{\mathcal{S}}$, we have

$$\left\| \left(M^\flat\right)\left(M^\flat\right)^\top \right\|_{\mathcal{S}} = \sup_{\substack{A^\flat \in \mathcal{S} \\ \|A\|_F = 1}} \left(A^\flat\right)^\top \left(M^\flat\right)\left(M^\flat\right)^\top A^\flat = \sup_{\substack{A \in \mathcal{S} \\ \|A\|_F = 1}} \langle A, M\rangle^2 .$$

By self duality of the Frobenius norm, we know that

$$\langle A, M\rangle = \langle A, M - I\rangle \leq \|M - I\|_F ,$$

since $I^\flat \in \mathcal{S}^\perp$. The result now follows. □

Proof of Theorem 2.2.15. Throughout this proof, let $w \in S_{n,\varepsilon}$ be the input to the separation oracle, and let $\widehat{\Sigma} = \widehat{\Sigma}(w) = \sum_{i=1}^n w_i X_i X_i^\top$. Let us first prove completeness. Observe that by Theorem 2.1.15, we know that restricted to $\mathcal{S}$, we have that $M_4 = 2I$. Therefore, by (2.13) we will not output a hyperplane in line 7. Moreover, by (2.12), we will not output a hyperplane in line 8. This proves completeness.

Thus it suffices to show soundness. Suppose that $w \notin \mathcal{C}_{c\gamma}$. We will make use of the following elementary fact:

Fact 2.2.20. *Let $A = \Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}$ and $B = \widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2}$. Then*

$$\|A^{-1} - I\|_F = \|B - I\|_F$$

Proof. In particular $A^{-1} = \Sigma^{1/2}\widehat{\Sigma}^{-1}\Sigma^{1/2}$. Using this expression and the fact that all the matrices involved are symmetric, we can write

$$\begin{aligned}
\|A^{-1} - I\|_F^2 &= \operatorname{tr}\left((A^{-1} - I)^\top (A^{-1} - I)\right) \\
&= \operatorname{tr}\left(\Sigma^{1/2}\widehat{\Sigma}^{-1}\Sigma\widehat{\Sigma}^{-1}\Sigma^{1/2} - 2\Sigma^{1/2}\widehat{\Sigma}^{-1}\Sigma^{1/2} - I\right) \\
&= \operatorname{tr}\left(\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1}\Sigma\widehat{\Sigma}^{-1/2} - 2\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2} - I\right) \\
&= \operatorname{tr}\left((B - I)^\top (B - I)\right) = \|B - I\|_F^2
\end{aligned}$$

where in the third line we have used the fact that the trace of a product of matrices is preserved under cyclic shifts. □

This allows us to show:

Claim 2.2.21. *Assume (2.11) holds with $\gamma_1 \leq O(\gamma)$ and assume furthermore that $\|A - I\|_F \geq c\gamma$. Then, if we let $\gamma' = \frac{(1-\varepsilon)c}{2}\gamma = \Theta(\gamma)$, we have*

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Z_i - w_b I^\flat \right\|_{\mathcal{S}} + \left\| \sum_{i \in S_{\text{bad}}} w_i Z_i - w_b I^\flat \right\|_{\mathcal{S}^\perp} \geq \gamma' \, . \tag{2.14}$$

Proof. Let A, B be as in Fact 2.2.20. Combining Lemma 2.2.17 and Fact 2.2.20 we have

$$\|A - I\|_F \geq c\gamma \Rightarrow \|B - I\|_F \geq \frac{c\gamma}{2} \, . \tag{2.15}$$

We can rewrite (2.11) as the expression $\sum_{i \in S_{\text{good}}} w_i X_i X_i^\top = w_g \Sigma^{1/2}(I + R)\Sigma^{1/2}$ where R is symmetric and satisfies $\|R\|_F \leq \gamma_1$. By the definition of $\widehat{\Sigma}$ we have that $\sum_{i=1}^N w_i Y_i Y_i^\top = I$, and so

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Y_i Y_i^\top - w_b I \right\|_F = \left\| \sum_{i \in S_{\text{good}}} w_i Y_i Y_i^\top - w_g I \right\|_F = w_g \left\| \widehat{\Sigma}^{-1/2} \Sigma^{1/2} (I + R) \Sigma^{1/2} \widehat{\Sigma}^{-1/2} - I \right\|_F$$

Furthermore we have

$$\left\| \widehat{\Sigma}^{-1/2} \Sigma^{1/2} R \Sigma^{1/2} \widehat{\Sigma}^{-1/2} \right\|_F \leq \gamma_1 \left\| \widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} \right\|_2 \, ,$$

by applying Lemma 2.2.18. And putting it all together we have

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Y_i Y_i^\top - w_b I \right\|_F \geq w_g \left(\left\| \widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} - I \right\|_F - \gamma_1 \left\| \widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} \right\|_2 \right)$$

It is easily verified that for $c > 10$, we have that for all γ, if $\|\widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} - I\|_F \geq c\gamma$, then

$$\|\widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} - I\|_F \geq 2\gamma \|\widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2}\|_2 \, .$$

Hence all this implies that

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Y_i Y_i^\top - w_b I \right\|_F \geq \gamma' ,$$

where $\gamma' = \frac{c(1-\varepsilon)}{2}\gamma = \Theta(\gamma)$. The desired result then follows from the Pythagorean theorem. □

Claim 2.2.21 tells us that if $w \notin C_{c\gamma}$, we know that one of the terms in (2.15) must be at least $\frac{1}{2}\gamma'$. We first show that if the first term is large, then the algorithm outputs a separating hyperplane:

Claim 2.2.22. *Assume that (2.11)-(2.13) hold with $\gamma_1, \gamma_2 \leq O(\gamma)$ and $\gamma_3 \leq O(\gamma \log 1/\varepsilon)$. Moreover, suppose that*

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Z_i - w_b I^\flat \right\|_{\mathcal{S}} \geq \frac{1}{2}\gamma' .$$

Then the algorithm outputs a hyperplane in line 7, and moreover, it is a separating hyperplane.

Proof. Let us first show that given these conditions, then the algorithm indeed outputs a hyperplane in line 7. Since $I^\flat \in S^\perp$, the first term is just equal to $\left\| \sum_{i \in S_{\text{bad}}} w_i Z_i \right\|_S$. But this implies that there is some $M^\flat \in S$ such that $\|M^\flat\|_2 = \|M\|_F = 1$ and such that

$$\sum_{i \in S_{\text{bad}}} w_i \langle M^\flat, Z_i \rangle \geq \frac{1}{2}\gamma' ,$$

which implies that

$$\sum_{i \in S_{\text{bad}}} \frac{w_i}{w_b} \langle M^\flat, Z_i \rangle \geq \frac{1}{2} \frac{\gamma'}{w_b} .$$

The w_i/w_b are a set of weights satisfying the conditions of Claim 2.2.16 and so this implies that

$$\begin{aligned}\sum_{i\in S_{\text{bad}}} w_i \langle M^\flat, Z_i\rangle^2 &\geq O\left(\frac{\gamma'^2}{w_b}\right) \\ &\geq O\left(\frac{\gamma'^2}{\varepsilon}\right) \qquad (2.16)\end{aligned}$$

Let $\widetilde{\Sigma} = \widehat{\Sigma}^{-1}\Sigma$. By Theorem 2.1.15 and (2.13), we have that

$$\sum_{i\in S_{\text{good}}} w_i Z_i Z_i^\top = w_g \left(\left(\widetilde{\Sigma}^\flat\right)\left(\widetilde{\Sigma}^\flat\right)^\top + 2\widetilde{\Sigma}^{\otimes 2} + \left(\widetilde{\Sigma}^{1/2}\right)^{\otimes 2} R \left(\widetilde{\Sigma}^{1/2}\right)^{\otimes 2} \right) ,$$

where $\|R\|_2 \leq \gamma_3$. Hence,

$$\begin{aligned}\left\| \sum_{i\in S_{\text{good}}} w_i Z_i Z_i^\top - 2I \right\|_S &= w_g \left\| \left(\widetilde{\Sigma}^\flat\right)\left(\widetilde{\Sigma}^\flat\right)^\top + 2\left(\widetilde{\Sigma}^{\otimes 2} - I\right) + (1-w_g)I + \left(\widetilde{\Sigma}^{1/2}\right)^{\otimes 2} R \left(\widetilde{\Sigma}^{1/2}\right)^{\otimes 2} \right\|_S \\ &\leq \|\widetilde{\Sigma} - I\|_F^2 + 2\|\widetilde{\Sigma} - I\|_2 + (1-w_g) + \|R\|\|\widetilde{\Sigma}\|^2 \\ &\leq 3\|\widetilde{\Sigma} - I\|_F^2 + \gamma\|\widetilde{\Sigma}\|^2 + O(\varepsilon) \; . \\ &\leq O\left(\gamma'^2 + \gamma'\right) \; , \qquad (2.17)\end{aligned}$$

since it is easily verified that $\gamma\|\widetilde{\Sigma}\|^2 \leq O(\|\widetilde{\Sigma} - I\|_F)$ as long as $\|\widetilde{\Sigma} - I\|_F \geq \Omega(\gamma)$, which it is by (2.15).

Equations 2.16 and 2.17 then together imply that

$$\sum_{i=1}^n w_i (M^\flat)^\top Z_i Z_i^\top (M^\flat) - (M^\flat)^\top I M^\flat \geq O\left(\frac{\gamma^2}{\varepsilon}\right) \; ,$$

and so the top eigenvalue of M is greater in magnitude than λ, and so the algorithm will output a hyperplane in line 7. Letting ℓ denote the hyperplane output by the algorithm, by the same calculation as for (2.17), we must have $\ell(w^*) < 0$, so this is indeed a separating hyperplane. Hence in this case, the algorithm correctly operates.

□

Moreover, observe that from the calculations in (2.17), we know that if we ever output a hyperplane in line 7, which implies that $\lambda \geq \Omega(\varepsilon \log^2 1/\varepsilon)$, then we must have that $\ell(w^*) < 0$.

Now let us assume that the first term on the LHS is less than $\frac{1}{2}\gamma'$, such that the algorithm does not necessarily output a hyperplane in line 7. Thus, the second term on the LHS of Equation 2.14 is at least $\frac{1}{2}\gamma'$. We now show that this implies that this implies that the algorithm will output a separating hyperplane in line 9.

Claim 2.2.23. *Assume that (2.11)-(2.13) hold. Moreover, suppose that*

$$\left\| \sum_{i \in S_{\text{bad}}} w_i Z_i - w_b I^\flat \right\|_{\mathcal{S}^\perp} \geq \frac{1}{2}\gamma' \ .$$

Then the algorithm outputs a hyperplane in line 9, and moreover, it is a separating hyperplane.

Proof. By the definition of $\mathcal{S}^\perp$, the assumption implies that

$$\left| \sum_{i \in S_{\text{bad}}} w_i \frac{\operatorname{tr}(Z_i^\sharp)}{\sqrt{d}} - M_b \sqrt{d} \right| \geq \frac{1}{2}\gamma' \ ,$$

which is equivalent to the condition that

$$\xi \sum_{i \in S_{\text{bad}}} w_i \left(\frac{\|Y_i\|_2^2}{\sqrt{d}} - \sqrt{d} \right) \geq \frac{(1-\varepsilon)\gamma'}{2} \ ,$$

for some $\xi \in \{-1, 1\}$. In particular, the algorithm will output a hyperplane

$$\ell(w) = \xi \sum_{i \in S} w_i \left(\frac{\|Y_i\|_2^2}{\sqrt{d}} - \sqrt{d} \right) - \lambda$$

in Step 9, where S is some set of size at most εn, and $\lambda = O(\gamma')$. Since it will not affect anything, for without loss of generality let us assume that $\xi = 1$. The other case is symmetrical.

It now suffices to show that $\ell(w^*) < 0$ always. Let $T = S \cap S_{\text{good}}$. By (2.12), we know that

$$\sum_{i \in T} \frac{1}{|T|} Y_i Y_i^\top - I = \widetilde{\Sigma}^{1/2} \left(I + A \right) \widetilde{\Sigma}^{1/2} - I \ ,$$

where $\|A\|_F = O\left(\gamma \frac{n}{|T|} \right)$. Hence,

$$\begin{aligned} \left\| \sum_{i \in T} \frac{1}{(1-\varepsilon)n} Y_i Y_i^\top - \frac{|T|}{(1-\varepsilon)n} I \right\|_F &= \frac{|T|}{(1-\varepsilon)n} \left\| \widetilde{\Sigma}^{1/2} \left(I + A \right) \widetilde{\Sigma}^{1/2} - I \right\|_F \\ &\leq \frac{|T|}{(1-\varepsilon)n} \left(\| \widetilde{\Sigma} - I \|_F + \|A\|_F \|\widetilde{\Sigma}\|_2 \right) \\ &\leq \frac{|T|}{(1-\varepsilon)n} \| \widetilde{\Sigma} - I \|_F + O(\gamma) \|\widetilde{\Sigma}\|_2 \\ &\leq O(\gamma\gamma' + \gamma) \ , \end{aligned}$$

as long as $\gamma' \geq O(\gamma)$. By self-duality of the Frobenius norm, using the test matrix $\frac{1}{\sqrt{d}} I$, this implies that

$$\left| \sum_{i \in T} \frac{1}{(1-\varepsilon)n} \left(\|Y_i\|^2 - \sqrt{d} \right) \right| \leq O(\gamma\gamma' + \gamma) < \alpha$$

and hence $\ell(w^*) < 0$, as claimed. □

These two claims in conjunction directly imply the correctness of the theorem. □

The full algorithm

As before, this separation oracle and Corollary 2.1.3 shows that we have demonstrated an algorithm FindApproxCovariance with the following properties:

Theorem 2.2.24. *Fix $\varepsilon, \delta > 0$, and let $\gamma = O(\varepsilon \log 1/\varepsilon)$. Let $c > 0$ be a universal constant which is sufficiently large. Let $X_1, \ldots, X_n$ be an ε-corrupted set of points satisfying (2.11-(2.13), for $\gamma_1, \gamma_2 \leq O(\gamma)$ and $\gamma_3 \leq O(\gamma \log 1/\varepsilon)$. Then* FindApproxCovariance$(\varepsilon, \delta, X_1, \ldots, X_n)$ *runs in time* $\mathrm{poly}(n, d, 1/\varepsilon, \log 1/\delta)$*, and outputs a u such that there is some $w \in C_{c\gamma}$ such that $\|w - u\|_\infty \leq \varepsilon/(nd \log(n/\delta))$.*

As before, this is not quite sufficient to actually recover the covariance robustly. Naively, we would just like to output $\sum_{i=1}^n u_i X_i X_i^\top$. However, this can run into issues if there are points X_i such that $\|\Sigma^{-1/2} X_i\|_2$ is extremely large. We show here that we can postprocess the u such that we can weed out these points. First, observe that we have the following lemma:

Lemma 2.2.25. *Assume $X_1, \ldots, X_n$ satisfy (2.11). Let $w \in S_{n,\varepsilon}$. Then*

$$\sum_{i=1}^n w_i X_i X_i^\top \succeq (1 - O(\gamma_1))\Sigma \; .$$

Proof. This follows since by (2.11), we have that $\sum_{i \in S_{\mathrm{good}}} w_i X_i X_i^\top \succeq w_g(1-\gamma_1)\Sigma \succeq (1 - O(\gamma_1))\Sigma$. The lemma then follows since $\sum_{i \in S_{\mathrm{bad}}} w_i X_i X_i^\top \succeq 0$ always. □

Now, for any set of weights $w \in S_{n,\varepsilon}$, let $\widetilde{w}^- \in \mathbb{R}^n$ be the vector given by $\widetilde{w}_i^- = \max(0, w_i - \varepsilon/(nd \log(n/\delta)))$, and let w^- be the set of weights given by renormalizing $\widetilde{w}^-$. It is a straightforward calculation that for any $w \in S_{n,\varepsilon}$, we have $w^- \in S_{n,2\varepsilon}$. In particular, this implies:

Lemma 2.2.26. *Let u be such that there is $w \in C_{c\gamma}$ such that $\|u - w\|_\infty \leq \varepsilon/(nd \log(n/\delta))$. Then, $\sum_{i=1}^n u_i^- X_i X_i^\top \preceq (1 + O(\gamma))\Sigma$.*

Proof. By the definition of $C_{c\gamma}$, we must have that $\sum_{i=1}^N w_i X_i X_i^\top \preceq (1 + c\gamma)\Sigma$. Moreover, we must have $\widetilde{u}_i^- \leq w_i$ for every index $i \in [n]$. Thus we have that $\sum_{i=1}^n \widetilde{u}_i^- w_i X_i X_i^\top \preceq (1 + c\gamma)\Sigma$, and hence $\sum_{i=1}^N u_i^- w_i X_i X_i^\top \preceq (1 + c\gamma)\Sigma$, since $\sum_{i=1}^N u_i^- w_i X_i X_i^\top \preceq (1 + O(\varepsilon)) \sum_{i=1}^N \widetilde{u}_i^- w_i X_i X_i^\top$. □

We now give the full algorithm. The algorithm proceeds as follows: first run FindApproxCovariance to get some set of weights u which is close to some element of $C_{c\gamma}$. We then compute the empirical covariance $\Sigma_1 = \sum_{i=1}^n u_i X_i X_i^\top$ with the weights u, and remove any points which have $\|\Sigma_1^{-1/2} X_i\|_2^2$ which are too large. We shall show that this removes no good points, and removes all corrupted points which have $\|\Sigma^{-1/2} X_i\|_2^2$ which are absurdly large. We then rerun FindApproxCovariance with this pruned set of points, and output the empirical covariance with the output of this second run. Formally, we give the pseudocode for the algorithm in Algorithm 5.

Algorithm 5 Full algorithm for learning the covariance agnostically

```
function LearnCovariance($\varepsilon, \delta, X_1, \ldots, X_N$)
    Let $u \leftarrow$ FindApproxCovariance($\varepsilon, \delta, X_1, \ldots, X_n$).
    Let $\Sigma_1 = \sum_{i=1}^n u_i^- X_i X_i^\top$.
    for $i = 1, \ldots, n$ do
        if $\|\Sigma_1^{-1/2} X_i\|_2^2 \geq \Omega(d \log N/\delta)$ then
            Remove $X_i$ from the set of samples
    Let $S'$ be the set of pruned samples.
    Let $u' \leftarrow$ FindApproxCovariance($\varepsilon, \delta, \{X_i\}_{i \in S'}$).
    return $\sum_{i=1}^n u'_i X_i X_i^\top$.
```

We now show that this algorithm is correct.

Theorem 2.2.27. *Let $1/2 \geq \varepsilon > 0$, and $\delta > 0$. Let $\gamma = O(\varepsilon \log 1/\varepsilon)$. Let $X_1, \ldots, X_n$ be a ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$ where*

$$n = \widetilde{\Omega}\left(\frac{d^2 \log^5 1/\delta}{\varepsilon^2}\right) .$$

Let $\widehat{\Sigma}$ be the output of LearnCovariance($\varepsilon, \delta, X_1, \ldots, X_n$). *Then with probability $1 - \delta$, $\|\Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} - I\|_F \leq O(\gamma)$.*

Proof. We first condition on the event that we satisfy (2.10)-(2.13) with $\gamma_1, \gamma_2 \leq O(\gamma)$ and $\gamma_3 \leq O(\gamma \log 1/\varepsilon)$. By our choice of n, Fact 2.1.11, Corollary 2.1.12, Corollary 2.1.14, and Theorem 2.1.16, and a union bound, we know that this event happens with probability $1 - \delta$.

By Theorem 2.2.24, Lemma 2.2.25, and Lemma 2.2.26, we have that since ε is sufficiently small,

$$\frac{1}{2}\Sigma \preceq \Sigma_1 \preceq 2\Sigma \ .$$

In particular, this implies that for every vector X_i, we have

$$\frac{1}{2}\|\Sigma^{-1/2} X_i\|_2^2 \leq \|\Sigma_1^{-1/2} X_i\|_2^2 \leq 2\|\Sigma^{-1/2} X_i\|_2^2 \ .$$

Therefore, by (2.10), we know that in line 6, we never throw out any uncorrupted points, and moreover, if X_i is corrupted with $\|\Sigma^{-1/2}X_i\|_2^2 \geq \Omega(d \log N/\delta)$, then it is thrown out. Thus, let S' be the set of pruned points. Because no uncorrupted point is thrown out, we have that $|S'| \geq (1-2\varepsilon)N$, and moreover, this set of points still satisfies (2.11)-(2.13)[2] and moreover, for ever $i \in S'$, we have $\|\Sigma^{-1/2}X_i\|_2^2 \leq O(d \log N/\delta)$. Therefore, by Theorem 2.2.24, we have that there is some $u'' \in C_{c|I|}$ such that $\|u' - u''\|_\infty < \varepsilon/(Nd\log(N/\delta))$. But now if $\widehat{\Sigma} = \sum_{i \in |I|} u'_i X_i X_i^\top$, we have

$$\begin{aligned}\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I\|_F &\leq \left\| \sum_{i \in I} u''_i \Sigma^{-1/2} X_i X_i^\top \Sigma^{-1/2} - I \right\|_F + \sum_{i \in I} |u'_i - u'_i| \|\Sigma^{-1/2} X_i\|_2^2 \\ &\leq c\gamma + O(\varepsilon) \leq O(\gamma) \ ,\end{aligned}$$

which completes the proof. □

2.2.4 Learning an arbitrary Gaussian agnostically

We have shown how to agnostically learn the mean of a Gaussian with known covariance, and we have shown how to agnostically learn the covariance of a mean zero Gaussian. In this section, we show how to use these two in conjunction to agnostically learn an arbitrary Gaussian. Throughout, let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$, where both μ and Σ are unknown. For the sake of simplicity assume that n is even.

For each $i = 1, \ldots, n/2$, let $X'_i = (X_i - X_{n/2+i})/\sqrt{2}$. Observe that if both X_i and $X_{n/2+i}$ are uncorrupted, then $X'_i \sim \mathcal{N}(0, \Sigma)$. Let ${S_{\mathrm{good}}}' \subseteq [n/2]$ denote the set of $i \in [n/2]$ so that both $i \in S_{\mathrm{good}}$ and $i + n/2 \in S_{\mathrm{good}}$, and let ${S_{\mathrm{bad}}}' = [n/2] \setminus {S_{\mathrm{good}}}'$, and let $w'_g = \sum_{i \in {S_{\mathrm{good}}}'} w_i$ and $w'_b = \sum_{i \in {S_{\mathrm{bad}}}'} w_i$. Thus observe that X'_i are an 2ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$ of size $n/2$. In analogy with Section 2.2.3, let $U'_i = \Sigma^{-1/2} X'_i$ and let $Z'_i = (U'_i)^{\otimes 2}$.

Our algorithm will work under the following set of deterministic conditions over

[2] Technically, the samples satisfy a slightly different set of conditions since we may have thrown out some corrupted points, and so in particular the number of samples may have changed, but the meaning should be clear.

the X_i and the X_i':

$$\|U_i'\|_2^2 \leq O\left(d \log(n/\delta)\right) \ , \forall i \in {S_{\text{good}}}' \tag{2.18}$$

$$\left\| \sum_{i \in {S_{\text{good}}}'} w_i'(U_i')(U_i')^\top - w_g' I \right\|_F \leq \gamma_1 \ , \tag{2.19}$$

$$\left\| \sum_{i \in T} \frac{1}{|T|}(U_i')(U_i')^\top - I \right\|_F \leq O\left(\gamma_2 \frac{n}{|T|} \right) \ , \text{and} \tag{2.20}$$

$$\left\| \sum_{i \in {S_{\text{good}}}'} w_i'(Z_i')(Z_i')^\top - w_g' M_4 \right\|_{\mathcal{S}} \leq \gamma_3 \tag{2.21}$$

$$\|\Sigma^{-1/2}(X_i - \mu)\Sigma^{-1/2}\|_2 \leq O\left(\sqrt{d \log(n/\delta)}\right), \forall i \in S_{\text{good}} \ , \tag{2.22}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i(X_i - \mu)(X_i - \mu)^\top - w_g I \right\|_\Sigma \leq \gamma_4 \ \forall w \in S_{n,2\varepsilon}, \text{ and} \tag{2.23}$$

$$\left\| \Sigma^{-1/2} \sum_{i \in S_{\text{good}}} w_i(X_i - \mu)\Sigma^{-1/2} \right\|_2 \leq \gamma_5 \ \forall w \in S_{n,2\varepsilon} \ . \tag{2.24}$$

for all $w' \in S_{n/2,2\varepsilon}$, and all sets $T \subseteq {S_{\text{good}}}'$ of size $|T| \leq \varepsilon n$. Here $\gamma_1, \ldots, \gamma_5$ are parameters, where we will set:

$$\gamma_1, \gamma_2, \gamma_4 \gamma_5 = O(\varepsilon \log(1/\varepsilon)) \ , \text{ and}$$
$$\gamma_3 = O(\varepsilon \log^2 1/\varepsilon) \ .$$

By applying the appropriate concentration inequalities and one massive union bound[3], with these settings of parameters one can check that all of these hold simultaneously with probability $1 - \delta$ so long as

$$n = \widetilde{\Omega}\left(\frac{d^2 \log^5 1/\delta}{\varepsilon^2} \right) \ .$$

[3] I'm sorry but I really don't want to go through and find the reference for all 7 (!) of these; the interested reader can find the original reference in the original section

From unknown mean, unknown covariance, to zero Mean, unknown covariance

Because the X'_i are a 2ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$, by using the results from Section 2.2.3, under conditions (2.18)-(2.13), we can recover $\widehat{\Sigma}$ so that

$$\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I\|_F \leq O(\gamma) \ , \tag{2.25}$$

where $\gamma = O(\varepsilon \log 1/\varepsilon)$.

From unknown Mean, approximate covariance, to approximate recovery

For each X_i, let $X''_i = \widehat{\Sigma}^{-1/2} X_i$. Then, it is readily seen that (2.22)-(2.24) imply that

$$\|(X''_i - \mu)\|_2 \leq O\left(\sqrt{d \log(n/\delta)}\right), \forall i \in S_{\text{good}} \ , \tag{2.26}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i (X''_i - \mu)(X''_i - \mu)^\top - w_g I \right\|_2 \leq \gamma_4 + O(\gamma) \ \forall w \in S_{n,2\varepsilon}, \text{ and} \tag{2.27}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i (X''_i - \mu) \right\|_2 \leq \gamma_5 + O(\gamma) \ \forall w \in S_{n,2\varepsilon} \ . \tag{2.28}$$

Let $\gamma' = O(\varepsilon\sqrt{\log 1/\varepsilon}) + \gamma = O(\varepsilon \log 1/\varepsilon)$. Then, by using results from Section 2.2.1, we can recover a $\widehat{\mu}$ such that $\|\widehat{\mu} - \widehat{\Sigma}^{-1/2}\mu\|_2 \leq O(\gamma')$. Observe here we are tacitly using the fact that our algorithms are additively tolerant to spectral noise.

From Parametric Recovery to TV Recovery To briefly recap, we now have obtained parameters $\widehat{\mu}, \widehat{\Sigma}$ so that:

$$\|\widehat{\mu} - \widehat{\Sigma}^{-1/2}\mu\|_2 \leq O(\gamma') \text{ , and } \|\widehat{\Sigma} - \Sigma\|_\Sigma \leq O(\gamma) \ ,$$

where $\gamma, \gamma' = O(\varepsilon \log 1/\varepsilon)$. We wish to show that these guarantees imply recovery in statistical distance.

First, by Fact 1.4.3, we have

$$d_{\mathrm{TV}}(\mathcal{N}(\widehat{\mu}, I), \mathcal{N}(\widehat{\Sigma}^{-1/2}\mu, I)) \leq O(\varepsilon \log(1/\varepsilon)) \ ,$$

or since TV distance is affine invariant,

$$d_{\mathrm{TV}}(\mathcal{N}(\widehat{\Sigma}^{1/2}\widehat{\mu}, \widehat{\Sigma}), \mathcal{N}(\mu, \widehat{\Sigma})) \leq O(\varepsilon \log(1/\varepsilon)) \ ,$$

which in conjunction with Corollary 1.4.6 and a triangle inequality, implies that

$$d_{\mathrm{TV}}(\mathcal{N}(\widehat{\Sigma}^{1/2}\widehat{\mu}, \widehat{\Sigma}), \mathcal{N}(\mu, \Sigma)) \leq O(\varepsilon \log(1/\varepsilon)) \ ,$$

and thus by following this procedure, whose formal pseudocode is given in Algorithm 6, we have shown the following:

Algorithm 6 Algorithm for learning an arbitrary Gaussian robustly

```
function RecoverRobustGuassian($\varepsilon, \tau, X_1, \ldots, X_N$)
    For $i = 1, \ldots, N/2$, let $X'_i = (X_i - X_{N/2+i})/\sqrt{2}$.
    Let $\widehat{\Sigma} \leftarrow$ LearnCovariance($\varepsilon, \tau, X'_1, \ldots, X'_{N/2}$).
    For $i = 1, \ldots, N$, let $X''_i = \widehat{\Sigma}^{-1/2} X_i$.
    Let $\widehat{\mu} \leftarrow$ LearnMean($\varepsilon, \tau, X''_1, \ldots, X''_N$).
    return the Gaussian with mean $\widehat{\Sigma}^{1/2}\widehat{\mu}$, and covariance $\widehat{\Sigma}$.
```

Theorem 2.2.28. *Fix $\varepsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$, where μ, Σ are both unknown, and*

$$n = \widetilde{\Omega}\left(\frac{d^2 \log^5 1/\delta}{\varepsilon^2}\right) \ .$$

There is a polynomial-time algorithm RecoverRobustGaussian$(\varepsilon, \tau, X_1, \ldots, X_n)$ *which with probability $1 - \delta$, outputs a $\widehat{\Sigma}, \widehat{\mu}$ such that*

$$d_{\mathrm{TV}}(\mathcal{N}(\widehat{\Sigma}^{1/2}\widehat{\mu}, \widehat{\Sigma}), \mathcal{N}(\mu, \Sigma)) \leq O(\varepsilon \log(1/\varepsilon)) \ .$$

Chapter 3

Convex Programming II: Robust Learning With Sparsity

Under the Moon we stand — at last
At Seasons change — where will we be?
I long to hear your Voice once more —
I did not know — in those days past
How Fortune was so kind to me —

3.1 Robust estimation in other norms

We view the mean estimation results presented in this chapter as a specific case of a more general phenomena in robust estimation. At a high level, the main technical work in this chapter is to learn the mean of a Gaussian in a "sparsity-inducing" norm, rather than ℓ_2. It turns out that for this specific norm, efficient robust mean estimation is possible, albeit at statistical cost. This begs the following natural question:

In what norms is robust mean estimation possible?

More specifically, we leave the following as a very interesting open question: given a class of distributions $\mathcal{D}$, and a norm $\| \cdot \|$, estimate $f(\varepsilon)$, where

$$f(\varepsilon) = \min_{\widehat{\mu}} \max_{\substack{D,D': \\ D \in \mathcal{D}, d_{\mathrm{TV}}(D,D') \leq \varepsilon}} \mathop{\mathbb{E}}_{X_1, \ldots, X_n \sim D} \left[\| \mu - \widehat{\mu}(X_1, \ldots, X_n) \| \right] ,$$

where the minimum is taken over all estimators $\widehat{\mu}$. In other words, give a simple characterization of what is the best rate achievable by any estimator for robust mean estimation of $\mathcal{D}$ in $\| \cdot \|$. Another interesting (and much harder) question is to characterize the best rate achievable for any efficient algorithm.

3.1.1 Generalizing spectral signatures

As a first attempt to understand this problem, let's understand what the natural generalization of spectral signatures are to any norm. Given a norm $\| \cdot \|$, recall that the dual norm of $\| \cdot \|$, denoted $\| \cdot \|^*$, is defined to be

$$\|u\|^* = \sup_{\|v\|=1} \langle u, v \rangle ,$$

and recall that $(\| \cdot \|^*)^* = \| \cdot \|$. Now let's phrase the intuition of spectral signatures in terms of this language, for more general norms.

Suppose we have a distribution D with mean μ, and an ε-corrupted data set $X_1, \ldots, X_n$ from D with empirical mean $\widehat{\mu}$. Now suppose that $\|\mu - \widehat{\mu}\| > \gamma(\varepsilon)$ is large. This means that there exists some dual vector v with $\|v\|^* = 1$ so that $\langle v, \mu - \widehat{\mu} \rangle > \gamma(\varepsilon)$. Expanding slightly, we have

$$\langle v, \mu - \widehat{\mu} \rangle = \frac{|S_{\mathrm{good}}|}{n} \left\langle v, \frac{1}{S_{\mathrm{good}}} \sum_{i \in S_{\mathrm{good}}} X_i - \mu \right\rangle + \frac{|S_{\mathrm{bad}}|}{n} \left\langle v, \frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} X_i - \mu \right\rangle . \tag{3.1}$$

Now suppose that we have concentration of the uncorrupted points to the mean, that is,

$$\left\| \frac{1}{S_{\mathrm{good}}} \sum_{i \in S_{\mathrm{good}}} (X_i - \mu) \right\| < O(\gamma(\varepsilon)) . \tag{3.2}$$

This implies that

$$\frac{|S_{\mathrm{bad}}|}{n} \left\langle v, \frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} X_i - \mu \right\rangle > \Omega\left(\gamma(\varepsilon)\right) \ , \tag{3.3}$$

so since $|S_{\mathrm{bad}}|/n = \varepsilon$, we have

$$\frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} \langle v, X_i - \mu \rangle > \Omega\left(\frac{\gamma(\varepsilon)}{\varepsilon}\right) \ .$$

Notice that this implies that

$$\begin{aligned} \frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} \langle v, X_i - \widehat{\mu} \rangle &= \frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} \langle v, X_i - \mu \rangle + \langle v, \mu - \widehat{\mu} \rangle \\ &> \Omega\left(\frac{\gamma(\varepsilon)}{\varepsilon}\right) + \langle v, \mu - \widehat{\mu} \rangle \\ &\geq \Omega\left(\frac{\gamma(\varepsilon)}{\varepsilon}\right) + \|\mu - \widehat{\mu}\| \\ &> \Omega\left(\frac{\gamma(\varepsilon)}{\varepsilon}\right) \ . \end{aligned}$$

Thus by Jensen's inequality we have

$$\frac{1}{|S_{\mathrm{bad}}|} \sum_{i \in S_{\mathrm{bad}}} \langle v, X_i - \widehat{\mu} \rangle^2 > \Omega\left(\frac{\gamma(\varepsilon)}{\varepsilon}\right)^2 \ .$$

Thus, this implies that

$$\max_{\|v\|^* = 1} v^\top \widehat{\Sigma} v > \Omega\left(\frac{\gamma(\varepsilon)^2}{\varepsilon}\right) \ . \tag{3.4}$$

Equation 3.4 is the natural generalization of spectral signatures to general norms. Our derivation above says that if every subset of size $(1-\varepsilon)n$ of the uncorrupted points converges to error $\gamma(\varepsilon)$ in $\|\cdot\|$, then the presence of outliers induces a deviation in the second moment of the order in Equation 3.4. Thus, if a deviation in the second moment of this order cannot happen without corruption, then this gives us a way to detect if the mean is being corrupted.

However, the main problem with Equation 3.4 is that this maximization problem is in general difficult to compute. When $\|\cdot\| = \|\cdot\|_2$ we were lucky because spectral methods suffice. Thus, to adapt the machinery of spectral signatures to efficient robust learning in general norms seems to require a non-trivial amount of problem-specific thought. We leave it as an interesting open question to give a systematic way of doing so. In the remainder of this chapter, we will show how we do so to solve robust mean estimation under sparsity. It will turn out that the similar, but more involved, ideas also allow us to attack robust sparse PCA.

3.2 Robust sparse estimation

In the last couple of decades, there has been a large amount of work in machine learning and statistics on how to exploit sparsity in high dimensional data analysis. Motivated by the ever-increasing quantity and dimensionality of data, the goal at a high level is to utilize the underlying sparsity of natural data to extract meaningful guarantees using a number of samples that is sublinear in the dimensionality of the data. In this chapter, we will consider the unsupervised setting, where we have sample access to some distribution with some underlying sparsity, and our goal is to recover this distribution by exploiting this structure. Two natural and well-studied problems in this setting that attempt to exploit sparsity are sparse mean estimation and sparse PCA. In both problems, the shared theme is that we assume that one wishes to find a distinguished sparse direction of a Gaussian data set. However, the algorithms inspired by this line of work tend to be quite brittle—it can be shown that they fail when the model is slightly perturbed.

This raises the natural "meta-question":

Question 3.2.1. Do the statistical gains (achievable by computationally efficient algorithms) for sparse estimation problems persist in the presence of noise?

More formally: Suppose we are asked to solve some estimation task given samples from some distribution D with some underlying sparsity constraint (e.g. sparse PCA). Suppose now an ε-fraction of the samples are corrupted. Can we still solve the same sparse estimation problem? Understanding this question—in a couple of fundamental settings—is the main focus of this chapter.

Interestingly, new gaps between computational and statistical rates seem to emerge in the presence of noise. In particular, while the sparse mean estimation problem was previously quite simple to solve, the efficient algorithms which achieve the minimax rate for this problem break down in the presence of this adversarial noise. More concretely, it seems that the efficient algorithms which are robust to noise run into the same computational issues as those which plague sparse PCA. A very interesting question is whether this phenomenon is inherent to any computationally efficient algorithm.

3.2.1 Our contribution

We study the natural robust versions of two classical, well-studied statistical tasks involving sparsity, namely, sparse mean estimation, and sparse PCA.

Robust sparse mean estimation Here, we get a set of d-dimensional samples from $\mathcal{N}(\mu, I)$, where μ is k-sparse, and an ε-fraction of the points are corrupted adversarially. Our goal then is to recover μ. Our main contribution is the following:

Theorem 3.2.2 (informal, see Theorem 3.3.1)**.** *There is an efficient algorithm, which given a set of ε-corrupted samples of size $\widetilde{O}(\frac{k^2 \log d}{\varepsilon^2})$ from $\mathcal{N}(\mu, I)$ where μ is k-sparse, outputs a $\widehat{\mu}$ so that with high probability, $\|\widehat{\mu} - \mu\|_2 \leq \varepsilon\sqrt{\log 1/\varepsilon}$.*

The recovery guarantee we achieve, namely $O(\varepsilon\sqrt{\log 1/\varepsilon})$, is off by the optimal guarantee by only a factor of $\sqrt{\log 1/\varepsilon}$. Moreover, results of [DKS16d] imply that our bound is tight for any efficient SQ algorithm. One can show that information theoretically, it suffices to take $O(\frac{k \log d}{\varepsilon^2})$ samples to learn the mean to ℓ_2 error $O(\varepsilon)$, even with corrupted data. Without model misspecification, this problem is quite simple algorithmically: it turns out that the truncated empirical mean achieves the information theoretically optimal rate. However, efficient algorithms for this task break down badly given noise, and to our knowledge there is no simple way of fixing them. Very interestingly, the rate we achieve is off from this information theoretic rate by a k^2 vs k factor—the same computational vs. statistical gap that arises in sparse PCA. This phenomenon only seems to appear in the presence of noise, and we conjecture that this is inherent:

Conjecture 3.2.1. Any efficient algorithm for robust sparse mean estimation needs $\widetilde{\Omega}(\frac{k^2 \log d}{\varepsilon^2})$ samples.

In Appendix C.3 we give some intuition for why it seems to be true. At a high level, it seems that any technique to detect outliers for the mean must look for sparse directions in which the variance is much larger than it should be; at which point the problem faces the same computational difficulties as sparse PCA. We leave closing this gap as an interesting open problem.

Robust sparse PCA Here, we study the natural robust analogue of the spiked covariance model. Classically, two problems are studied in this setting. The *detection* problem is given as follows: given sample access to the distributions, we are asked to distinguish between $\mathcal{N}(0, I)$, and $\mathcal{N}(0, I + \rho v v^\top)$ where v is a k-sparse unit vector. That is, we wish to understand if we can detect the presence of any sparse principal component. Our main result is the following:

Theorem 3.2.3 (informal, see Theorem 3.3.2)**.** *Fix $\rho > 0$, and let $\eta = O(\varepsilon\sqrt{\log 1/\varepsilon})$. If $\rho > \eta$, there is an efficient algorithm, which given a set of ε-corrupted samples of size $O(\frac{k^2 \log d}{\rho^2})$ which distinguishes between $\mathcal{N}(0, I)$, and $\mathcal{N}(0, I + \rho v v^\top)$ with high probability.*

The condition that $\varepsilon = \widetilde{O}(\rho)$ is necessary (up to log factors), as otherwise the problem is impossible information theoretically. Observe that this (up to log factors) matches the optimal rate for computationally efficient detection for sparse PCA without noise (under reasonable complexity theoretic assumptions, see [BR13, WBS16]), and so it seems that noise does not introduce an additional gap here. The *recovery* problem is similar, except now we want to recover the planted spike v, i.e. find a u

minimizing

$$L(u,v) = \frac{1}{\sqrt{2}} \left\| uu^\top - vv^\top \right\|_2 \ , \tag{3.5}$$

which turns out to be the natural measure for this problem. For this, we show:

Theorem 3.2.4 (informal, see Theorem 3.3.3)**.** *Fix $\varepsilon > 0$ and $0 < \rho = O(1)$, and let $\eta = O(\varepsilon \sqrt{\log 1/\varepsilon})$. There is an efficient algorithm, which given a set of ε-corrupted samples of size $O(\frac{k^2 \log d}{\eta^2})$ from $\mathcal{N}(0, I + \rho vv^\top)$, outputs a u so that $L(u,v) = O\left(\frac{\eta}{\rho}\right)$ with high probability.*

This rate is non-trivial—in particular, it provides guarantees for recovery of v when the number of samples we take is at the detection threshold. Moreover, up to log factors, our rate is optimal for computationally efficient algorithms–[WBS16] gives an algorithm with rate roughly $O(\varepsilon/\rho)$, and show that this is necessary.

Techniques We first introduce a simple way to describe the optimization problems used for solving sparse mean estimation and sparse PCA. This approach is very similar to the approach taken by [CRPW12] for solving under-determined linear systems. We observe that any set $\mathcal{S}$ in a Hilbert space naturally induces a dual norm $\|x\|_{\mathcal{S}}^* = \max_{y \in \mathcal{S}} |\langle x, y \rangle|$, and that well-known efficient algorithms for sparse mean estimation and sparse PCA simply compute this norm, and the corresponding dual witness $y \in \mathcal{S}$ which maximizes this norm, for appropriate choices of $\mathcal{S}$. These norms give us a language to only consider deviations in directions we care about, which allows us to prove concentration bounds which are not true for more traditional norms.

We now describe our techniques for robust sparse mean estimation. Our starting point is the convex programming approach of Chapter 2. We assign each sample point a weight, which morally corresponds to our belief about whether the point is corrupted, and we optimize these weights. In the previous chapter, the approach was to find weights so that the empirical covariance with these weights looked like the identity in spectral norm.

Unfortunately, such an approach fundamentally fails for us because the spectrum of the covariance will never concentrate for us with the number of samples we take.

Instead, we utilize a novel connection to sparse PCA. We show that if instead we find weights so that the empirical covariance with these weights looks like the identity in the dual norm induced by a natural SDP for sparse PCA (in the noiseless setting), then this suffices to show that the truncated empirical mean with these weights is close to the truth.

Essentially, by robustly learning the mean in another norm (i.e. not Euclidean norm) which respects sparsity, we are able to recover the mean. This is where the connection to robust learning in general norms comes in. Ideally, if we could follow the procedure given above exactly, then we could almost exactly recover the same statistical guarantees for this problem with the same number of samples, as if there were no adversarial noise. However, the maximization problem that directly arises from following the procedure, while information theoretically sufficient, is computationally difficult.

To circumvent this, we relax the maximization problem. We show that in fact the dual norm induced by the SDP for sparse PCA gives a reasonable proxy for it, and that this dual norm maximization problem can be solved efficiently, albeit at a slight cost in the number of samples. This in turns suffices to allow us to (approximately) find a point in the desired feasible set of points, which we show suffices to recover the true mean.

We now turn to robust sparse PCA. We first consider the detection problem, which is somewhat easier technically. Here, we again use the dual norm induced by the SDP for sparse PCA. We show that if we can find weights on the samples (as before) so that the empirical covariance with these samples has minimal dual norm, then the value of the dual norm gives us a distinguisher between the spiked and non-spiked case. To find such a set of weights, we observe that norms are convex, and thus our objective is convex. Thus, as before, to optimize over this set it suffices to give a separation oracle, which again the SDP for sparse PCA allows us to do.

We now turn our attention to the recovery problem. Here, the setup is very similar, except now we simultaneously find a set of weights and an "explainer" matrix A so that the empirical covariance with these weights is "maximally explained" by

A, in a norm very similar to the one induced by the sparse PCA SDP. Utilizing that norms are convex, we show that this can be done via a convex program using the types of techniques described above, and that the top eigenvector of the optimal A gives us the desired solution. While the convex program would be quite difficult to write down in one shot, it is quite easily expressible using the abstraction of dual norms.

3.2.2 Related work

As mentioned previously, there has been a large amount of work on various ways to exploit sparsity for machine learning and statistics. In the supervised setting, perhaps the most well-known of these is compressive sensing and its variants (see [CW08, HTW15] for more details). We do not attempt to provide an exhaustive overview the field here. Other well-known problems in the same vein include general classes of linear inverse problems, see [CRPW12] and matrix completion ([CR09]).

The question of estimating a sparse mean is very related to a classical statistical model known as the *Gaussian sequence model*, and the reader is referred to [Tsy08, Joh13, RH17] for in-depth surveys on the area. This problem has also garnered a lot of attention recently in various distributed and memory-limited settings, see [GMN14, SD15, BGM$^+$16]. The study of sparse PCA was initiated in [Joh01] and since yielded a very rich algorithmic and statistical theory ([dEGJL07, dBG08, AW09, WTH09, JNRS10, ACCD11, LZ12, Ma13, BJNP13, CMW$^+$13, OMH$^+$14, GWL14, CRZ$^+$16, PWBM16, BMV$^+$18]). In particular, we highlight a very interesting line of work [BR13, KNV$^+$15, MW15, WGL16, WBS16, HKP$^+$17], which give evidence that any computationally efficient estimator for sparse PCA must suffer a sub-optimal statistical rate rate. We conjecture that a similar phenomenon occurs when we inject noise into the sparse mean estimation problem.

3.3 Definitions

We will now formally define the algorithmic problems we consider.

Robust sparse mean estimation Here, we assume we get an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$, where μ is k-sparse. Our goal is to recover μ in ℓ_2. It is not hard to show that there is an exponential time estimator which achieves rate $\widetilde{O}(k \log d / \varepsilon^2)$, and moreover, this rate is optimal (see Appendix C.1). However, this algorithm requires highly exponential time. We show:

Theorem 3.3.1 (Efficient robust sparse mean estimation)**.** *Fix $\varepsilon, \delta > 0$, and let k be fixed. Let $\eta = O(\varepsilon \sqrt{\log 1/\varepsilon})$. Given an ε-corrupted set of samples $X_1, \ldots, X_n \in \mathbb{R}^d$ from $\mathcal{N}(\mu, I)$, where μ is k-sparse, and*

$$n = \Omega\left(\frac{\min(k^2, d) + \log \binom{d^2}{k^2} + \log 1/\delta}{\eta^2} \right) ,$$

there is a poly-time algorithm which outputs $\widehat{\mu}$ so that w.p. $1 - \delta$, we have $\|\mu - \widehat{\mu}\|_2 \leq O(\eta)$.

It is well-known that information theoretically, the best error one can achieve is $\Theta(\varepsilon)$, as achieved by Fact C.1.1. We show that it is possible to efficiently match this bound, up to a $\sqrt{\log 1/\varepsilon}$ factor. Interestingly, our rate differs from that in Fact C.1.1: our sample complexity is (roughly) $\widetilde{O}(k^2 \log d / \varepsilon^2)$ versus $O(k \log d / \varepsilon^2)$. We conjecture this is necessary for any efficient algorithm.

Robust sparse PCA We will consider both the detection and recovery problems for sparse PCA. We first focus *detection problem* for sparse PCA. Here, we are given a signal-to-noise ratio (SNR, see e.g. [Twi17]) $\rho > 0$, and an ε-corrupted set of samples from a d-dimensional distribution D, where D can is either $\mathcal{N}(0, I)$ or $\mathcal{N}(0, I + \rho v v^\top)$ for some k-sparse unit vector v. Our goal is to distinguish between the two cases, using as few samples as possible. It is not hard to show that information theoretically, $O(k \log d / \rho^2)$ samples suffice for this problem, with an inefficient algorithm (see Appendix C.1). Our first result is that efficient robust sparse PCA detection is possible, at effectively the best computationally efficient rate:

Theorem 3.3.2 (Robust sparse PCA detection)**.** *Fix $\rho, \delta, \varepsilon > 0$. Let $\eta = O(\varepsilon \sqrt{\log 1/\varepsilon})$. Then, if $\eta = O(\rho)$, and we are given a we are given a ε-corrupted set of samples from* *either $\mathcal{N}(0, I)$ or $\mathcal{N}(0, I + \rho vv^\top)$ for some k-sparse unit vector v of size*

$$n = \Omega\left(\frac{\min(d, k^2) + \log\binom{d^2}{k^2} + \log 1/\delta}{\rho^2}\right)$$

then there is a polynomial time algorithm which succeeds with probability $1 - \delta$ for detection.

It was shown in [BR13] that even without noise, at least $n = \Omega(k^2 \log d/\varepsilon^2)$ samples are required for any polynomial time algorithm for detection, under reasonable complexity theoretic assumptions. Up to log factors, we recover this rate, even in the presence of noise.

We next consider the *recovery* problem. Here, we are given an ε-corrupted set of samples from $\mathcal{N}(0, I + \rho vv^\top)$, and our goal is to output a u minimizing $L(u, v)$, where $L(u, v) = \frac{1}{\sqrt{2}}\|uu^\top - vv^\top\|_2$. For the recovery problem, we recover the following efficient rate:

Theorem 3.3.3 (Robust sparse PCA recovery)**.** *Fix $\varepsilon, \rho > 0$. Let η be as in Theorem 3.3.2. There is an efficient algorithm, which given a set of ε-corrupted samples of size n from $\mathcal{N}(0, I + \rho vv^\top)$, where*

$$n = \Omega\left(\frac{\min(d, k^2) + \log\binom{d^2}{k^2} + \log 1/\delta}{\eta^2}\right) ,$$

outputs a u so that

$$L(u, v) = O\left(\frac{(1 + \rho)\eta}{\rho}\right) .$$

In particular, observe that when $\eta = O(\rho)$, so when $\varepsilon = \widetilde{O}(\rho)$, this implies that we recover v to some small constant error. Therefore, given the same number of samples as in Theorem 3.3.2, this algorithm begins to provide non-trivial recovery guarantees. Thus, this algorithm has the right "phase transition" for when it begins to work, as this number of samples is likely necessary for any computationally efficient algorithm. Moreover, our rate itself is likely optimal (up to log factors), when $\rho = O(1)$. In the non-robust setting, [WBS16] showed a rate of (roughly) $O(\varepsilon/\rho)$ with the same number of samples, and that any computationally efficient algorithm cannot beat this rate. We leave it as an interesting open problem to show if this rate is achievable or not in the presence of error when $\rho = \omega(1)$.

3.4 Concentration for sparse estimation problems via dual norms

In this section we give a clean way of proving concentration bounds for various objects which arise in sparse PCA and sparse mean estimation problems. We do so by observing they are instances of a very general "meta-algorithm" we call dual norm maximization. This will prove crucial to proving the correctness of our algorithms for robust sparse recovery. While this may sound similar to the "dual certificate" techniques often used in the sparse estimation literature, these techniques are actually quite different.

Definition 3.4.1 (Dual norm maximization)**.** Let $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$. Fix any set $S \subseteq \mathcal{H}$. Then the dual norm induced by S, denoted $\| \cdot \|_S^*$, is defined by $\|x\|_S^* = \sup_{y \in S} |\langle x, y \rangle|$. The dual norm maximizer of x, denoted $d_S(x)$, is the vector $d_S(x) = \arg\max_{v \in S} |\langle v, x \rangle|$.

In particular, we will use the following two sets. Equip the space of symmetric $d \times d$ matrices with the trace inner product, i.e., $\langle A, B \rangle = \mathrm{tr}(AB)$, so that it is a Hilbert space, and let

$$\mathcal{U}_k = \{u \in \mathbb{R}^d : \|u\|_2 = 1, \|u\|_0 = k\} \tag{3.6}$$

$$\mathcal{X}_k = \{X \in \mathbb{R}^{d \times d} : \mathrm{tr}(X) = 1, \|X\|_1 \leq k, X \succeq 0\} \ . \tag{3.7}$$

We show in Appendix C.2.1 that existing well-known algorithms for sparse mean recovery and sparse PCA without noise can be naturally written in this fashion.

Another detail we will largely ignore in this paper is the fact that efficient algorithms for these problems can only approximately solve the dual norm maximization problem. However, we explain in Appendix C.2.2 why this does not affect us in any meaningful way. Thus, for the rest of the paper we will assume we have access to the exact maximizer, and the exact value of the norm.

3.4.1 Concentration for dual norm maximization

We now show how to derive very strong concentration results for the dual norm maximization problem for $\mathcal{U}_k$ and $\mathcal{X}_k$. Conceptually, we view these concentration results as being the major distinction between sparse estimation and non-sparse estimation tasks. Indeed, these results are crucial for adapting the convex programming framework for robust estimation to sparse estimation tasks. Additionally, they allow us to give an easy proof that the L_1 relaxation works for sparse PCA.

Corollary 3.4.1. *Let n be a positive integer, and let $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$. Then there are universal constants $A, B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^n X_i\right\|_{\mathcal{U}_k}^* > t\right] \leq 4\exp\left(A\left(k + \log\binom{d}{k}\right) - Bnt^2\right) .$$

Proof. Fix a set of k coordinates $S \subseteq [d]$, and let V_S denote the space of unit vectors supported on S. By Lemma 2.1.4 and a net argument, we have that

$$\Pr\left[\exists v \in V_S : \left|\left\langle v, \frac{1}{n}\sum_{i=1}^n X_i\right\rangle\right| > t\right] \leq 4\exp\left(Ak - Bnt^2\right) .$$

with probability $1 - \delta$. The result then follows by further union bounding over all $\binom{d}{k}$ sets of k coordinates. □

The second concentration bound, which bounds deviation in $\mathcal{X}_k$ norm, uses ideas which are similar at a high level, but requires a bit more technical work.

Theorem 3.4.2. *Let n be a positive integer, and let $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$. Then* *there are universal constants $A, B > 0$ so that for all $t > 0$, we have*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^{n} X_i X_i^\top - I\right\|_{\mathcal{X}_k}^{*} > t\right] \leq 4\exp\left(A\left(\min(d,k^2) + \log\binom{d^2}{k^2}\right) - Bn\min(t,t^2)\right) .$$

Let us first introduce the following definition.

Definition 3.4.2. A *symmetric sparsity pattern* is a set S of indices $(i,j) \in [d] \times [d]$ so that if $(i,j) \in S$ then $(j,i) \in S$. We say that a symmetric matrix $M \in \mathbb{R}^{d\times d}$ respects a symmetric sparsity pattern S if $\mathrm{supp}(M) = S$.

We also let $\mathcal{A}_k$ denote the set of symmetric matrices $M \in \mathbb{R}^{d\times d}$ with $\|M\|_0 \leq k^2$ and $\|M\|_F \leq 1$. With these definition, we now show:

Lemma 3.4.3. *For all $t > 0$, we have*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^{n} X_i X_i^\top - I\right\|_{\mathcal{A}_k}^{*} > t\right]$$
$$\leq 4\exp\left(A\left(\min(d,k^2) + \log\binom{d^2}{k^2}\right) - Bn\min(t,t^2)\right) .$$

Proof. Fix any symmetric sparsity pattern S so that $|S| \leq k^2$. By classical arguments one can show that there is a $(1/3)$-net over all symmetric matrices X with $\|X\|_F = 1$ respecting S of size at most $9^{O(\min(d,k^2))}$. By Lemma 2.1.5 and a basic net argument, we know that

$$\Pr\left[\exists M \in \mathcal{W}_k \text{ s.t. } \mathrm{supp}(M) = S : \left|\frac{1}{n}\sum_{i=1}^{n} \left\langle M, X_i X_i^\top\right\rangle - \langle M, I\rangle\right| > t\right]$$
$$\leq 4\exp\left(A\min(d,k^2) - Bn\min(t,t^2)\right) .$$

The claim then follows by further union bounding over all $O\left(\binom{d^2}{k^2}\right)$ symmetric sparsity patterns S with $|S| \leq k^2$. □

We will also require the following structural lemma.

Lemma 3.4.4. *Any positive semi-definite matrix $X \in \mathbb{R}^{d \times d}$ so that $\mathrm{tr}(X) \leq 1$ and $\|X\|_1 \leq k$ can be written as*

$$X = \sum_{i=1}^{O(n^2/k^2)} Y_i \; ,$$

where each Y_i is symmetric, have $\sum_{i=1}^{O(n^2/k^2)} \|Y_i\|_F \leq 4$, and each Y_i is k^2-sparse.

Proof. Observe that since X is positive semi-definite, then $\|X\|_F \leq \mathrm{tr}(X) \leq 1$. For simplicity of exposition, let us ignore that the Y_i must be symmetric for this proof. We will briefly mention how to in addition ensure that the Y_i are symmetric at the end of the proof. Sort the entries of X in order of decreasing $|X_{ij}|$. Let Y_i be the matrix whose nonzeroes are the $ik^2 + 1$ through $(i+1)k^2$ largest entries of X, in the same positions as they appear in X. Then we clearly have that $\sum Y_i = X_i$, and each Y_i is exactly k^2-sparse.[1] Thus it suffices to show that $\sum \|Y_i\|_F \leq 4$. We have $\|Y_1\|_F \leq \|X\|_F \leq 1$. Additionally, we have $\|Y_{i+1}\|_F \leq \frac{1^\top |Y_i| 1}{k}$, which follows simply because every nonzero entry of Y_{i+1} is at most the smallest entry of Y_i, and each has exactly k^2 nonzeros (except potentially the last one, but it is not hard to see this cannot affect anything). Thus, in aggregate we have

$$\sum_{i=1}^{O(n^2/k^2)} \|Y_i\|_F \leq 1 + \sum_{i=2}^{O(n^2/k^2)} \|Y_i\|_F \leq 1 + \sum_{i=1}^{O(n^2/k^2)} \frac{1^\top |Y_i| 1}{k} = 1 + \frac{1^\top |X| 1}{k} \leq 2 \; ,$$

which is stronger than claimed.

However, as written it is not clear that the Y_i's must be symmetric, and indeed they do not have to be. The only real condition we needed was that the Y_i's (1) had disjoint support, (2) summed to X, (3) are each $\Theta(k^2)$ sparse (except potentially the last one), and (4) the largest entry of Y_{i+1} is bounded by the smallest entry of Y_i. It should be clear that this can be done while respecting symmetry by doubling the number of Y_i, which also at most doubles the bound in the sum of the Frobenius norms. We omit the details for simplicity. □

[1] Technically the last Y_i may not be k^2 sparse but this is easily dealt with, and we will ignore this case here

Proof of Theorem 3.4.2. We show that that for any symmetric matrix M, we have

$$\|M\|^*_{\mathcal{X}_k} \leq 4 \cdot \|M\|^*_{\mathcal{A}_k} .$$

Then, the desired conclusion follows from Lemma 3.4.3.

Indeed, by Lemma 3.4.4, for all $X \in \mathcal{X}_k$, we have that

$$X = \sum_{i=1}^{O(d^2/k^2)} Y_i ,$$

where each Y_i is symmetric, have $\sum_{i=1}^{O(d^2/k^2)} \|Y_i\|_F \leq 4$, and each Y_i is k^2-sparse. Thus,

$$\begin{aligned} |\langle X, M \rangle| &\leq \sum_{i=1}^{O(d^2/k^2)} |\langle Y_i, M \rangle| \\ &= \sum_{i=1}^{O(d^2/k^2)} \|Y_i\|_F \left| \left\langle \frac{Y_i}{\|Y_i\|_F}, M \right\rangle \right| \\ &\overset{(a)}{\leq} \sum_{i=1}^{O(d^2/k^2)} \|Y_i\|_F \|M\|^*_{\mathcal{W}_k} \\ &\overset{(b)}{\leq} 4 \cdot \|M\|^*_{\mathcal{A}_k} , \end{aligned}$$

where (a) follows since $Y_i/\|Y_i\|_F \in \mathcal{A}_k$, and (b) follows from the bound on the sum of the Frobenius norms of the Y_i. □

3.4.2 Concentration for $S_{n,\varepsilon}$

We will require the following concentration inequalities for weighted sums of Gaussians, where the weights come from $S_{n,\varepsilon}$, as these objects will naturally arise in our algorithms. These follow from the same union bound technique as used in Lemma 2.1.8, so we will omit the details of the proofs.

Theorem 3.4.5. *Fix $\varepsilon \leq 1/2$ and $\delta \leq 1$, and fix $k \leq d$. There is a $\eta_1 = O(\varepsilon\sqrt{\log 1/\varepsilon})$*

so that for any $\eta > \eta_1$, if $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$ and

$$n = \Omega\left(\frac{\min(d, k^2) + \log\binom{d^2}{k^2} + \log 1/\delta}{\eta^2}\right) ,$$

then

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\| \frac{1}{n}\sum_{i=1}^n w_i X_i \right\|_{\mathcal{U}_k}^* \geq \eta \right] \leq \delta .$$

Proof of Theorem 3.4.5. This follows from the exact same technique as the proof of Lemma 2.1.8, but using Corollary 3.4.1 rather than Lemma 2.1.4. □

Theorem 3.4.6. *Fix $\varepsilon \leq 1/2$ and $\delta \leq 1$, and fix $k \leq d$. There is a $\eta = O(\varepsilon \log 1/\varepsilon)$ so that if $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$ and*

$$n = \Omega\left(\frac{\min(d, k^2) + \log\binom{d^2}{k^2} + \log 1/\delta}{\eta^2}\right) ,$$

then we have

$$\Pr\left[\exists w \in S_{n,\varepsilon} : \left\| \frac{1}{n}\sum_{i=1}^n w_i X_i X_i^\top - I \right\|_{\mathcal{X}_k}^* \geq \eta \right] \leq \delta .$$

Again, this follows from the exact same techniques as the proof of Lemma 2.1.8, and using Theorem 3.4.2.

3.5 A robust algorithm for robust sparse mean estimation

This section is dedicated to the description of an algorithm RecoverRobustSMean for robustly learning Gaussian sequence models, and the proof of the following theorem:

Theorem 3.5.1. *Fix* $\varepsilon, ^\top > 0$. *Let* $\eta = O(\varepsilon\sqrt{\log 1/\varepsilon})$. *Given an* ε*-corrupted set of samples of size* n *from* $\mathcal{N}(\mu, I)$, *where* μ *is* k*-sparse*

$$n = \Omega\left(\frac{\min(k^2, d) + \log\binom{d^2}{k^2} + \log 1/\delta}{\eta^2}\right) ,$$

then RecoverRobustSMean *outputs a* $\widehat{\mu}$ *so that with probability* $1 - \delta$, *we have* $\|\widehat{\mu} - \mu\|_2 \leq O(\eta)$.

Our algorithm builds upon the convex programming framework developed in the previous chapter. Roughly speaking, the algorithm proceeds as follows. First, it does a simple naive pruning step to remove all points which are more than roughly $\Omega(\sqrt{d})$ away from the mean. Then, for an appropriate choice of δ, it will attempt to (approximately) find a point within the following convex set:

$$C_\tau = \left\{ w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^{n} w_i (X_i - \mu)(X_i - \mu)^\top - I \right\|_{\mathcal{X}_k}^{*} \leq \tau \right\} . \tag{3.8}$$

The main difficulty with finding a point in C_τ is that μ is unknown. Recall that a key insight from Chapter 2 is that it suffices to create an (approximate) separation oracle for the feasible set, as then we may use classical convex optimization algorithms (i.e. ellipsoid or cutting plane methods) to find a feasible point. In their setting (for a different C_τ), it turns out that a simple spectral algorithm suffices to give such a separation oracle.

Our main contribution is the design of separation oracle for C_τ, which requires more sophisticated techniques. In particular, we will ideas developed in analogy to hard thresholding and SDPs similar to those developed for sparse PCA to design such an oracle.

3.5.1 Deterministic conditions

Throughout this section, we will condition on the following three deterministic events occurring:

$$\textsc{NaivePrune}(X_1, \ldots, X_n, \delta) \text{ succeeds,} \tag{3.9}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i (X_i - \mu) \right\|_{\mathcal{U}_k}^* \leq \eta_1 \ , \ \forall w \in S_{n, 2\varepsilon} \ , \text{ and} \tag{3.10}$$

$$\left\| \sum_{i \in S_{\text{good}}} w_i (X_i - \mu)(X_i - \mu)^\top - w_g I \right\|_{\mathcal{X}_k}^* \leq \eta_2 \ , \ \forall w \in S_{n, 2\varepsilon} \ , \tag{3.11}$$

where

$$\eta_1 := O(\varepsilon \sqrt{\log 1/\varepsilon}) \text{ and } \eta_2 := O(\varepsilon \log 1/\varepsilon) \ . \tag{3.12}$$

Let $\eta := \max(\eta_1, \eta_2)$. When $n = \Omega\left(\frac{\min(k^2, d) + \log \binom{k^2}{d^2} + \log 1/\delta}{\eta^2} \right)$ these events simultaneously happen with probability at least $1 - O(\delta)$ by Fact 2.2.6, Theorem 3.4.5, Theorem 3.4.6 and a union bound, and the observation that if $w \in S_{n,\varepsilon}$, then w/w^g restricted to the indices in S_{good} is in $S_{(1-\varepsilon)n, 2\varepsilon}$.

3.5.2 The separation oracle

Our main result in this section is the description of a polynomial time algorithm RobustSMeanOracle and the proof of the following theorem of its correctness:

Theorem 3.5.2. *Fix $\varepsilon > 0$ sufficiently small. Suppose that (3.10) and (3.11) hold. Let w^* denote the set of weights which are uniform over the uncorrupted points. Then, there is a constant $1 \leq c \leq 21$ so that* RobustSMeanOracle *satisfies:*

1. *(Completeness) If $w = w^*$,* RobustSMeanOracle *outputs "YES".*

2. *(Soundness) If $w \notin C_{c\eta}$ the algorithm outputs a hyperplane $\ell : \mathbb{R}^n \to \mathbb{R}$ so that $\ell(w) \geq 0$ but $\ell(w^*) < 0$. Moreover, if the algorithm ever outputs a hyperplane, we have $\ell(w^*) < 0$.*

Plugging these guarantees into an ellipsoid (or cutting-plane) method e.g. as given in [GLS88], we obtain the following:

Corollary 3.5.3. *Fix $\varepsilon > 0$ sufficiently small. Suppose that (3.10) and (3.11) hold. There is an algorithm* ApproxRecoverRobustSMean *which queries* RobustSMeanOracle *at most* $\mathrm{poly}(d, 1/\varepsilon, \log 1/\delta)$ *times, and so runs in time* $\mathrm{poly}(d, 1/\varepsilon, 1/\delta)$ *which outputs a w' so that $\|w - w'\|_\infty \leq \varepsilon/(n\sqrt{d \log n/\delta})$, for some $w \in C_{c\tau}$.*

Our separation oracle, formally described in Algorithm 7, proceeds as follows. Given $w \in S_{n,\varepsilon}$, it forms $\widehat{\mu} = \widehat{\mu}(w) = \sum w_i X_i$. It then forms the matrix $\widehat{\Sigma} = \widehat{\Sigma}(w) = \sum w_i (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top$, and computes $A = d_{\mathcal{X}_k}(\widehat{\Sigma})$. The algorithm then checks if $\left|\langle A, \widehat{\Sigma} \rangle\right| > C$ for appropriately chosen threshold C. If it does not, the algorithm outputs "YES". Otherwise, the algorithm outputs a separating hyperplane given by this matrix A.

Algorithm 7 Separation oracle for robust sparse mean estimation.

```
function RobustSMeanOracle($X_1, \ldots, X_n$, $w$)
    Let $\widehat{\mu} = \sum w_i X_i$
    Let $\widehat{\Sigma} = \sum w_i (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top$
    Let $A = d_{\mathcal{X}_k}(\widehat{\Sigma})$
    if $|\langle A, \widehat{\Sigma} - I \rangle| \geq 20\eta_2$ then
        Let $\sigma = \mathrm{sgn}\left(\langle A, \widehat{\Sigma} - I \rangle\right)$
        return the hyperplane $\ell$ given by
            $$\ell(w) = \sigma \cdot \left( \sum_{i=1}^n w_i \left\langle A, (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top \right\rangle - 1 \right) - |\langle A, \widehat{\Sigma} - I \rangle| .$$
    else
        return "YES"
    end
```

We require the following lemma:

Lemma 3.5.4. *Let $u \in \mathbb{R}^d$. Then $(\|u\|^*_{\mathcal{U}_k})^2 \leq \|uu^\top\|^*_{\mathcal{X}_k} \leq 4(\|u\|^*_{\mathcal{U}_k})^2$.*

Proof. Let $v = d_{\mathcal{U}_k}(u)$. Then since $vv^\top \in \mathcal{X}_k$, we have that $(\|uu^\top\|^*_{\mathcal{X}_k}) \geq \langle vv^\top, uu^\top \rangle = \langle u, v \rangle^2 = (\|u\|^*_{\mathcal{U}_k})^2$. This proves the first inequality.

To prove the other inequality, we first prove the intermediate claim that

$$\sup_{M \in \mathcal{Y}_{k^2}} u^\top M u \leq (\|u\|^*_{\mathcal{U}_k})^2 ,$$

where $\mathcal{Y}_{k^2}$ is the set of symmetric matrices M with at most k^2-non-zeroes satisfying $\|M\|_F = 1$. Indeed, fix any $M \in \mathcal{Y}_k$. Let $S \subseteq [n]$ be the set of non-zeroes of $d_{\mathcal{U}_k}(u)$. This is exactly the set of the k largest elements in u, sorted by absolute value. Let P be the symmetric sparsity pattern respected by M. Fix an arbitrary bijection $\phi : P \setminus (S \times S) \to (S \times S) \setminus P$, and let M' be the following matrix:

$$M'_{i,j} = \begin{cases} M_{ij} & \text{if } (i,j) \in P \bigcap (S \times S) , \\ \operatorname{sgn}(u_i u_j) \, M_{\phi^{-1}(i,j)} & \text{if } (i,j) \in (S \times S) \setminus P , \\ 0 & \text{otherwise.} \end{cases}$$

Then we claim that $u^\top M u \leq u^\top M' u$. Indeed, we have

$$\begin{aligned} u^\top M' u - u^\top M u &= \sum_{(i,j) \in P \setminus (S \times S)} |M_{ij} (uu^\top)_{\phi(i,j)}| - M_{ij} (uu^\top)_{i,j} \\ &\geq \sum_{(i,j) \in P \setminus (S \times S)} |M_{i,j}| \left(|(uu^\top)_{\phi(i,j)}| - |(uu^\top)_{i,j}| \right) \geq 0 , \end{aligned}$$

from the definition of S. Moreover, for any M respecting $S \times S$ with $\|M\|_F = 1$, it is not hard to see that $u^\top M u \leq (\|u\|^*_{\mathcal{U}_k})^2$. This is because now the problem is equivalent to restricting our attention to the coordinates in S, and asking for the symmetric matrix $M \in \mathbb{R}^{S \times S}$ with $\|M\|_F = 1$ maximizing $u_S^\top M u_S$, where u_S is u restricted to the coordinates in S. This is clearly maximized by $M = \frac{1}{\|u_S\|_2^2} u_S u_S^\top$, which yields the desired expression, since $\|u_S\|_2 = \|u\|_{\mathcal{U}_k}$.

We can now prove the original lemma. By Lemma 3.4.4 we may write $A = \sum_{i=1}^{O(n^2/k^2)} Y_i$ where each Y_i is symmetric, k^2-sparse, and have $\sum_{i=1}^{O(n^2/k^2)} \|Y_i\|_F \leq 4$.

We therefore have

$$
\begin{aligned}
u^\top A u &= \sum_{i=1}^{O(n^2/k^2)} u^\top Y_i u \\
&= \sum_{i=1}^{O(n^2/k^2)} \|Y_i\|_F (\|u\|_{\mathcal{U}_k}^*)^2 \\
&\leq 4(\|u\|_{\mathcal{U}_k}^*)^2 ,
\end{aligned}
$$

as claimed, where the second line follows from the arguments above. □

Throughout the rest of this section, let $Y_i = X_i - \mu$, so that so that $Y_i \sim \mathcal{N}(0, I)$ if $i \in S_{\text{good}}$. We first prove the following crucial proposition:

Proposition 3.5.5. *Let $w \in S_{n,\varepsilon}$, and let $\tau \geq \eta_1$. Assuming (3.10) and (3.11) hold, if $\|\sum_{i=1}^n w_i Y_i\|_{\mathcal{U}_k}^* \geq 3\tau_1$, then $\left\|\sum_{i=1}^n w_i Y_i Y_i^\top - I\right\|_{\mathcal{X}_k}^* \geq \frac{\tau^2}{\varepsilon}$.*

Proof. Observe that (3.10) and a triangle inequality together imply that

$$
\left\| \sum_{i \in S_{\text{bad}}} w_i Y_i \right\|_{\mathcal{U}_k}^* \geq 2\tau \; .
$$

By definition, this implies there is a k-sparse unit vector u so that $\left|\langle u, \sum_{i \in S_{\text{bad}}} w_i Y_i \rangle\right| \geq 2\tau$. WLOG assume that $\langle u, \sum_{i \in S_{\text{bad}}} w_i Y_i \rangle \geq \eta$ (if the sign is negative a symmetric argument suffices). This is equivalent to the statement that

$$
\sum_{i \in S_{\text{bad}}} \frac{w_i}{w^b} \langle u, Y_i \rangle \geq \frac{2\tau}{w^b} \; .
$$

Observe that the w_i / w^b are a set of non-negative weights summing to 1. Hence, by Lemma 2.2.16, we have

$$
\sum_{i \in S_{\text{bad}}} \frac{w_i}{w^b} \langle u, Y_i \rangle^2 \geq \left(\frac{2\tau}{w^b} \right)^2 \; .
$$

Let $A = uu^\top$. Observe that $A \in \mathcal{X}_k$. Then the above inequality is equivalent to the statement that

$$\sum_{i \in S_{\mathrm{bad}}} w_i Y_i^\top A Y_i \geq \frac{\tau^2}{w^b} \geq \frac{4\tau^2}{\varepsilon} \, .$$

Moreover, by (3.11), we have

$$\left| \sum_{i \in S_{\mathrm{good}}} w_i Y_i^\top A Y_i - I \right| \leq \eta \, ,$$

and together these two inequalities imply that

$$\sum_{i=1}^n w_i Y_i A Y_i \geq \frac{4\tau^2}{\varepsilon} - \eta \geq \frac{\tau^2}{\varepsilon} \, ,$$

as claimed. The final inequality follows from the definition of η, and since $4 > 2$. □

Proof of Theorem 3.5.2. Completeness follows from (3.11). We will now show soundness. Suppose $w \notin C_{21\eta}$. We wish to show that we will output a separating hyperplane. From the description of the algorithm, this is equivalent to showing that $\|\widehat{\Sigma} - I\|_{\mathcal{X}_k} \geq 20\eta_2$. Let $\widehat{\mu} = \widehat{\mu}(w) = \sum_{i=1}^n w_i X_i$, and let $\Delta = \mu - \widehat{\mu}$. By elementary manipulations, we may write

$$\begin{aligned}
\left\| \sum_{i=1}^n w_i (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top - I \right\|_{\mathcal{X}_k} &= \left\| \sum_{i=1}^n w_i (Y_i + \Delta)(Y_i + \Delta)^\top - I \right\|_{\mathcal{X}_k} \\
&\overset{(a)}{=} \left\| \sum_{i=1}^n w_i Y_i Y_i^\top + \Delta\Delta^\top - I \right\|_{\mathcal{X}_k} \\
&\overset{(b)}{\geq} \left\| \sum_{i=1}^n w_i Y_i Y_i^\top - I \right\|_{\mathcal{X}_k} - \left\| \Delta\Delta^\top \right\|_{\mathcal{X}_k} \\
&\overset{(c)}{\geq} \left\| \sum_{i=1}^n w_i Y_i Y_i^\top - I \right\|_{\mathcal{X}_k} - 4 \left\| \Delta \right\|_{\mathcal{U}_k}^2 \, ,
\end{aligned}$$

where (a) follows since $\sum_{i=1}^n w_i Y_i = \Delta$ by definition, (b) follows from a triangle inequality, and (c) follows from Lemma 3.5.4. If $\|\Delta\|_{\mathcal{U}_k} \leq \sqrt{\eta_2}/2$, then the RHS is at least $20\eta_2$ since the second term is at most η_2, and the first term is at least 21η since we assume that $w \notin C_{21\eta}$. Conversely, if $\|\Delta\|_{\mathcal{U}_k} \geq \sqrt{\eta_2/2}$, then by Proposition 3.5.5, we have $\| \sum_{i=1}^n w_i Y_i Y_i - I\|_{\mathcal{X}_k} \geq \|\Delta\|_{\mathcal{X}_k}^2/(6\varepsilon) > 48\|\Delta\|_{\mathcal{X}_k}^2$ as long as $\varepsilon \leq 1/288$. This implies that the RHS is at least $40\|\Delta\|_{\mathcal{X}_k^2} \geq 20\eta$, as claimed.

Hence, this implies that if $w \notin C_{4\eta}$, then we output a hyperplane ℓ. It is clear by construction that $\ell(w) \geq 0$; thus, it suffices to show that if we output a hyperplane, that $\ell(w^*) < 0$. Letting $\widetilde{\mu} = \frac{1}{(1-\varepsilon)n} \sum_{i \in S_{\text{good}}} w_i Y_i$, we have

$$\begin{aligned}
\sum_{i=1}^n w_i^* (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top - I &= \frac{1}{(1-\varepsilon)n} \sum_{i \in S_{\text{good}}} (Y_i + \Delta)(Y_i + \Delta)^\top - I \\
&= \frac{1}{(1-\varepsilon)n} \left(\sum_{i \in S_{\text{good}}} Y_i Y_i^\top - I \right) + \Delta \widetilde{\mu}^\top + \widetilde{\mu} \Delta^\top + \Delta \Delta^\top \\
&= \frac{1}{(1-\varepsilon)n} \left(\sum_{i \in S_{\text{good}}} Y_i Y_i^\top - I \right) + (\Delta + \widetilde{\mu})(\Delta + \widetilde{\mu})^\top - \widetilde{\mu}\widetilde{\mu}^\top .
\end{aligned}$$

Hence by the triangle inequality and Lemma 3.5.4, we have

$$\begin{aligned}
\left\| \sum_{i=1}^n w_i^* (X_i - \widehat{\mu})(X_i - \widehat{\mu})^\top - I \right\|_{\mathcal{X}_k} &\leq \left\| \frac{1}{1(1-\varepsilon)n} \sum_{i \in S_{\text{good}}} Y_i Y_i^\top - I \right\|_{\mathcal{X}_k}^* \\
&\quad + 4 \left(\|\Delta + \widetilde{\mu}\|_{\mathcal{U}_k}^* \right)^2 + 4 \left(\|\widetilde{\mu}\|_{\mathcal{U}_k}^* \right)^2 \\
&\leq \left\| \frac{1}{1(1-\varepsilon)n} \sum_{i \in S_{\text{good}}} Y_i Y_i^\top - I \right\|_{\mathcal{X}_k} + 8 \left(\|\Delta\|_{\mathcal{U}_k}^* \right)^2 \\
&\quad + 8 \left(\|\widetilde{\mu}\|_{\mathcal{U}_k}^* \right)^2 + 4 \left(\|\widetilde{\mu}\|_{\mathcal{U}_k}^* \right)^2 \\
&\leq 13\eta_2 + 8 \left(\|\Delta\|_{\mathcal{U}_k}^* \right)^2 \ , \qquad (3.13)
\end{aligned}$$

by (3.10) and (3.11).

Observe that to show that $\ell(w^*) < 0$ it suffices to show that

$$\left\| \sum_{i=1}^n w_i^* (X_i - \widehat{\mu})(X_i - \widehat{\mu}) - I \right\|_{\mathcal{X}_k}^* < \left\| \widehat{\Sigma} - I \right\|_{\mathcal{X}_k}^* \ . \qquad (3.14)$$

If $\|\Delta\|_{\mathcal{U}_k}^* \leq \sqrt{\eta_2/2}$, then this follows since the quantity on the RHS is at least 20η

by assumption, and the quantity on the LHS is at most 17η by (3.13). If $\|\Delta\|^*_{\mathcal{U}_k} \geq \sqrt{\eta/2}$, then by Proposition 3.5.5, the RHS of (3.14) is at least $\left(\|\Delta\|^*_{\mathcal{U}_k}\right)^2/(3\varepsilon)$, which dominates the LHS as long as $\|\Delta\|^*_{\mathcal{U}_k} \geq \eta_1$ and $\varepsilon \leq 1/288$, which completes the proof. □

3.5.3 Putting it all together

We now have the ingredients to prove our main theorem. Given what we have, our full algorithm RecoverRobustSMean is straightforward: first run NaivePrune, then run ApproxRecoverRobustSMean on the pruned points to output some set of weights w. We then output $\|\widehat{\mu}\|_{\mathcal{U}_k} d_{\mathcal{U}_k}(\widehat{\mu})$. The algorithm is formally defined in Algorithm 8.

Algorithm 8 An efficient algorithm for robust sparse mean estimation

```
function RecoverRobustSMean($X_1, \ldots, X_n, \varepsilon, \delta$)
    Let $S$ be the set output by NaivePrune($X_1, \ldots, X_n, \delta$). WLOG assume $S = [n]$.
    Let $w' =$ ApproxRecoverRobustSMean($X_1, \ldots, X_n, \varepsilon, \delta$).
    Let $\widehat{\mu} = \sum_{i=1}^n w'_i X_i$.
    return $\|\widehat{\mu}\|^*_{\mathcal{U}_k} d_{\mathcal{U}_k}(\widehat{\mu})$
```

Proof of Theorem 3.5.1. Let us condition on the event that (3.9), (3.10), and (3.11) all hold simultaneously. As previously mentioned, when $n = \Omega\left(\frac{\min(k^2,d)+\log\binom{k^2}{d^2}+\log 1/\delta}{\eta^2}\right)$ these events simultaneously happen with probability at least $1 - O(\delta)$. For simplicity of exposition, let us assume that NaivePrune does not remove any points. This is okay since if it succeeds, it never removes any good points, so if it removes any points, it can only help us. Moreover, since it succeeds, we know that $\|X_i - \mu\|_2 \leq O(\sqrt{d \log(n/\delta)})$ for all $i \in [n]$. By Corollary 3.5.3, we know that there is some $w \in C_{21\eta}$ so that $\|w - w'\|_\infty \leq \varepsilon/(n\sqrt{d \log n/\delta})$. We have

$$\begin{aligned}\|\widehat{\mu} - \mu\|_{\mathcal{U}_k} = \left\|\sum_{i=1}^n w'_i X_i - \widehat{\mu}\right\|^*_{\mathcal{U}_k} &\leq \left\|\sum_{i=1}^n w_i X_i - \widehat{\mu}\right\|^*_{\mathcal{U}_k} + \sum_{i=1}^n |w_i - w'_i| \, \|X_i - \mu\|_2 \\ &\leq O(\eta) + O(\varepsilon) \ ,\end{aligned}$$

by Proposition 3.5.5. We now show that this implies that if we let $\mu' = \|\widehat{\mu}\|_{\mathcal{U}_k}^* d_{\mathcal{U}_k}(\widehat{\mu})$, then $\|\mu' - \mu\|_2 \leq O(\eta)$. Let S be the support of μ', and let T be the support of μ. Then we have

$$\|\mu' - \mu\|_2^2 = \sum_{i \in S \cap T} (\mu_i' - \mu_i)^2 + \sum_{i \in S \setminus T} (\mu_i')^2 + \sum_{i \in T \setminus S} \mu_i^2 \; .$$

Observe that $\sum_{i \in S \cap T} (\mu_i' - \mu_i)^2 + \sum_{i \in S \setminus T} (\mu_i')^2 \leq \left(\|\widehat{\mu} - \mu\|_{\mathcal{U}_k}^* \right)^2$, since μ was originally nonzero on the entries in $S \setminus T$. Moreover, for all $i \in T \setminus S$ and $j \in S \setminus T$, we have $(\mu_i')^2 \leq (\mu_j')^2$. Thus we have

$$\sum_{i \in T \setminus S} \mu_i^2 \leq 2 \left(\sum_{i \in T \setminus S} (\mu - \mu_i')^2 + \sum_{i \in S \setminus T} (\mu_j')^2 \right) \leq 2 \left(\|\widehat{\mu} - \mu\|_{\mathcal{U}_k}^* \right)^2 \; .$$

Therefore we have $\|\mu' - \mu\|_2^2 \leq 3 \left(\|\widehat{\mu} - \mu\|_{\mathcal{U}_k}^* \right)^2$, which implies that $\|\mu' - \mu\|_2 \leq O(\eta_1)$, as claimed. □

3.6 An algorithm for robust sparse PCA detection

In this section, we give an efficient algorithm for detecting a spiked covariance matrix in the presence of adversarial noise. Throughout this section, let $\eta = O(\varepsilon \log 1/\varepsilon)$ be as in (3.12), and let $\rho = O(\eta)$.

Our algorithm is fairly straightforward: we ask for the set of weights $w \in S_{n,\varepsilon}$ so that the empirical second moment with these weights has minimal deviation from the identity in the dual $\mathcal{X}_k$ norm. We may write this as a convex program. Then, we check the value of the optimal solution of this convex program. If this value is small, then we say it is $\mathcal{N}(0, I)$. if this value is large, then we say it is $\mathcal{N}(0, I + \rho v v^\top)$. We refer to the former as Case 1 and the latter as Case 2. The formal description of this algorithm is given in Algorithm 9.

Algorithm 9 Detecting a spiked covariance model, robustly

```
function DetectRobustSPCA($X_1, \dots, X_n, \varepsilon, \delta, \rho$)
    Let $\gamma$ be the value of the solution
        $\min_{w \in S_{n,\varepsilon}} \left\| \sum_{i=1}^{n} w_i (X_i X_i^\top - I) \right\|_{\mathcal{X}_k}^{*}$    (3.15)
    if $\gamma < \rho/2$ then return Case 1 else return Case 2
```

3.6.1 Implementing DetectRobustSPCA

We first show that the algorithm presented above can be efficiently implemented. Indeed, one can show that by taking the dual of the SDP defining the $\| \cdot \|_{\mathcal{X}_k}^*$ norm, this problem can be re-written as an SDP with (up to constant factor blowups) the same number of constraints and variables, and therefore we may solve it using traditional SDP solver techniques.

Alternatively, one may observe that to optimize Algorithm 10 via ellipsoid or cutting plane methods, it suffices to, given $w \in S_{n,\varepsilon}$, produce a separating hyperplane for the constraint (3.15). This is precisely what dual norm maximization allows us to do efficiently. It is straightforward to show that the volume of $S_{n,\varepsilon} \times \mathcal{X}_k$ is at most exponential in the relevant parameters. Therefore, by the classical theory of convex optimization, (see e.g. [GLS88]), for any ξ, we may find a solution w' and γ' so that $\|w' - w^*\|_\infty \leq \xi$ and γ' so that $|\gamma - \gamma'| < \xi$ for some exact minimizer w^*, where γ is the true value of the solution, in time $\mathrm{poly}(d, n, 1/\varepsilon, \log 1/\xi)$,

As mentioned in Section C.2.2, neither approach will in general give exact solutions, however, both can achieve inverse polynomial accuracy in the parameters in polynomial time. We will ignore these issues of numerical precision throughout the remainder of this section, and assume we work with exact γ.

Observe that in general it may be problematic that we don't have exact access to the minimizer w^*, since some of the X_i may be unboundedly large (in particular, if it's corrupted) in norm. However, we only use information about γ. Since γ lives within a bounded range, and our analysis is robust to small changes to γ, these numerical issues do not change anything in the analysis.

3.6.2 Proof of Theorem 3.3.2

We now show that Algorithm 10 provides the guarantees required for Theorem 3.3.2. We first show that if we are in Case 1, then γ is small:

Lemma 3.6.1. *Let $\rho, \delta > 0$. Let ε, η be as in Theorem 3.3.2. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(0, I)$ of size n, where n is as in Theorem 3.3.2. Then, with probability $1 - \delta$, we have $\gamma \leq \rho/2$.*

Proof. Let w be the uniform weights over the uncorrupted points. Then it from Theorem 3.4.2 that $\| \sum_w w_i (X_i X_i^\top - I) \|_{\mathcal{X}_k}^* \leq O(\eta)$ with probability $1 - \delta$. Since $w \in S_{n,\varepsilon}$, this immediately implies that $\gamma \leq O(\rho)$. By setting constants appropriately, we obtain the desired guarantee. □

We now show that if we are in Case 2, then γ must be large:

Lemma 3.6.2. *Let $\rho, \delta > 0$. Let ε, η, n be as in Theorem 3.3.2. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(0, I)$ of size n. Then, with probability $1 - \delta$, we have $\gamma \geq (1 - \varepsilon)\rho - (2 + \rho)\eta$. In particular, for ε sufficiently small, and $\eta = O(\rho)$, we have that $\gamma > \rho/2$.*

Proof. Let $\Sigma = I + \rho v v^\top$, and let $Y_i = \Sigma^{-1/2} X_i$, so that if Y_i is uncorrupted, then $Y_i \sim \mathcal{N}(0, I)$. Let w^* be the optimal solution to (3.15). By Theorem 3.4.6, we have that with probability $1 - \delta$, we can write $\sum_{i=1}^n w_i^* Y_i Y_i^\top = w^g (I + N) + B$, where $\|N\|_{\mathcal{X}_k}^* \leq \eta_2$, and $B = \sum_{i \in S_{\text{bad}}} w_i^* Y_i Y_i^\top$. Therefore, we have $\sum_{i=1}^n w^* X_i X_i^\top = w^g (\Sigma + \Sigma^{1/2} N \Sigma^{1/2}) + \Sigma^{1/2} B \Sigma^{1/2}$. By definition, we have

$$\begin{aligned}
\left\| \sum_{i=1}^n w_i^* (X_i X_i^\top - I) \right\|_{\mathcal{X}_k}^* &\geq \langle w^g (\Sigma + \Sigma^{1/2} N \Sigma^{1/2}) + \Sigma^{1/2} B \Sigma^{1/2} - I, v v^\top \rangle \\
&\geq w^g \langle (\Sigma + \Sigma^{1/2} N \Sigma^{1/2}), v v^\top \rangle - 1 \\
&= w^g (1 + \rho) + w^g v^\top \Sigma^{1/2} N \Sigma^{1/2} v - 1 \\
&\geq (1 - \varepsilon)\rho + (1 - \varepsilon) v^\top \Sigma^{1/2} N \Sigma^{1/2} v - \varepsilon \, .
\end{aligned}$$

It thus suffices to show that $|v^\top \Sigma^{1/2} N \Sigma^{1/2} v| < (1+\rho)\eta$. Since v is an eigenvector for Σ with eigenvalue $1+\rho$, we have that $\Sigma^{1/2} v = \sqrt{\rho+1} \cdot v$ and thus

$$v^\top \Sigma^{1/2} N \Sigma^{1/2} v = (1+\rho) v^\top N v = (1+\rho)\langle N, vv^\top \rangle \leq (1+\rho) \|N\|^*_{\mathcal{X}_k} \leq (1+\rho)\eta \; .$$

□

Lemmas 3.6.1 and 3.6.2 together imply the correctness of DetectRobustSPCA and Theorem 3.3.2.

3.7 An algorithm for robust sparse PCA recovery

In this section, we prove Theorem 3.3.3. As in the previous section, let $\eta = O(\varepsilon \log 1/\varepsilon)$ be as in (3.12), and let $\rho = O(\eta)$.

We give some intuition here. Perhaps the first naive try would be to simply run the same SDP in (3.15), and hope that the dual norm maximizer gives you enough information to recover the hidden spike. This would more or less correspond to the simplest modification SDP of the sparse PCA in the non-robust setting that one could hope gives non-trivial information in this setting. However, this cannot work, for the following straightforward reason: the value of the SDP is always at least $O(\rho)$, as we argued in Section 3.6. Therefore, the noise can pretend to be some other sparse vector u orthogonal to v, so that the covariance with noise looks like $w^g(I + \rho vv^\top) + w^g \rho uu^\top$, so that the value of the SDP can be minimized with the uniform set of weights. Then it is easily verified that both $vv^\top$ and $uu^\top$ are dual norm maximizers, and so the dual norm maximizer does not uniquely determine v.

To circumvent this, we simply add an additional slack variable to the SDP, which is an additional matrix in $\mathcal{X}_k$, which we use to try to maximally explain away the rank-one part of $I + \rho vv^\top$. This forces the value of the SDP to be very small, which allows us to show that the slack variable actually captures v.

3.7.1 The algorithm

Our algorithms and analyses will make crucial use of the following convex set, which is a further relaxation of $\mathcal{X}_k$:

$$\mathcal{W}_k = \left\{ X \in \mathbb{R}^{d \times d} : \mathrm{tr}(X) \leq 2, \|X\|_2 \leq 1, \|X\|_1 \leq 3k, X \succeq 0 \right\} .$$

Our algorithm, given formally in Algorithm 10, will be the following. We solve a convex program which simultaneously chooses a weights in $S_{n,\varepsilon}$ and a matrix $A \in \mathcal{W}_k$ to minimize the $\mathcal{W}_k$ distance between the sample covariance with these weights, and A. Our output is then just the top eigenvector of A.

Algorithm 10 Learning a spiked covariance model, robustly

```
function RecoverRobustSPCA($X_1, \ldots, X_n, \varepsilon, \delta, \rho$)
    Let $w^*, A^*$ be the solution to
        $\arg\min_{w \in S_{n,\varepsilon}, A \in \mathcal{X}_k} \left\| \sum_{i=1}^n w_i (X_i X_i^\top - I) - \rho A \right\|_{\mathcal{W}_{2k}}^*$    (3.16)
    Let $u$ be the top eigenector of $A^*$
    return The $d_{\mathcal{U}_k}(u) \|u\|_{\mathcal{U}_k}^*$, i.e., the vector with all but the top $k$ coordinates of $v$ zeroed out.
```

This algorithm can be run efficiently for the same reasons as explained for DetectRobustSPCA. For the rest of the section we will assume that we have an exact solution for this problem. As before, we only use information about A, and since A comes from a bounded space, and our analysis is robust to small perturbations in A, this does not change anything.

3.7.2 More concentration bounds

Before we can prove correctness of our algorithm, we require a couple of concentration inequalities for the set $\mathcal{W}_k$.

Lemma 3.7.1. *Let n be a positive integer. Let $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$. Then*

$$\begin{aligned}\Pr &\left[\left\| \frac{1}{n} \sum_{i=1}^{n} X_i X_i^\top - I \right\|_{\mathcal{W}_k}^{*} > t \right] \\ &\leq 4 \exp\left(A \left(\min(d, k^2) + \log \binom{d^2}{k^2} \right) - Bn \min(t, t^2) \right) \ .\end{aligned}$$

Proof. It suffices to show that for any symmetric matrix $M \in \mathbb{R}^{d \times d}$, we have $\|M\|_{\mathcal{W}_k}^* \leq C \cdot \|M\|_{\mathcal{A}_k}^*$, as then the desired conclusion follows from Lemma 3.4.3. The proof is identical to the proof of Theorem 3.4.2 given Lemma 3.4.3, so we omit it for clarity. □

By the same techniques as in the proofs of Theorems 3.4.5 and 3.4.6, we can show the following bound. Because of this, we omit the proof for conciseness.

Corollary 3.7.2. *Fix $\varepsilon, \delta > 0$. Let $X_1, \ldots, X_n \sim \mathcal{N}(0, I)$ where n is as in Theorem 3.4.6. Then there is an $\eta = O(\varepsilon \log 1/\varepsilon)$ so that*

$$\Pr \left[\exists w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^{n} w_i X_i X_i^\top - I \right\|_{\mathcal{W}_k}^{*} \geq \eta \right] \leq \delta \ .$$

3.7.3 Proof of Theorem 3.3.3

In the rest of this section we will condition on the following deterministic event happening:

$$\forall w \in S_{n,\varepsilon} : \left\| \sum_{i=1}^{n} w_i X_i X_i^\top - I \right\|_{\mathcal{W}_{2k}}^{*} \leq \eta \ , \tag{3.17}$$

where $\eta = O(\varepsilon \log 1/\varepsilon)$. By Corollary 3.7.2, this holds if we take

$$n = \Omega \left(\frac{\min(d, k^2) + \log \binom{d^2}{k^2} + \log 1/\delta}{\eta_2^2} \right)$$

samples.

The rest of this section is dedicated to the proof of the following theorem, which immediately implies Theorem 3.3.3.

Theorem 3.7.3. *Fix ε, δ, and let η be as in (3.17). Assume that (3.17) holds. Let $\widehat{v}$ be the output of* $\textsc{RecoveryRobustSPCA}(X_1, \ldots, X_n, \varepsilon, \delta, \rho)$. *Then $L(\widehat{v}, v) \leq O(\sqrt{(1+\rho)\eta/\rho})$.*

Our proof proceeds in a couple of steps. Let $\Sigma = I + \rho v v^\top$ denote the true covariance. We first need the following, technical lemma:

Lemma 3.7.4. *Let $M \in \mathcal{W}_k$. Then $\Sigma^{1/2} M \Sigma^{1/2} \in (1+\rho)\mathcal{W}_k$.*

Proof. Clearly, $\Sigma^{1/2} M \Sigma^{1/2} \succeq 0$. Moreover, since $\Sigma^{1/2} = I + (\sqrt{1+\rho} - 1) v v^\top$, we have that the maximum value of any element of $\Sigma^{1/2}$ is upper bounded by $\sqrt{1+\rho}$. Thus, we have $\|\Sigma^{1/2} M \Sigma^{1/2}\|_1 \leq (1+\rho)\|M\|_1$. We also have

$$\begin{aligned} \operatorname{tr}(\Sigma^{1/2} M \Sigma^{1/2}) &= \operatorname{tr}(\Sigma M) \\ &= \operatorname{tr}(M) + \rho v^\top M v \leq 1 + \rho \ , \end{aligned}$$

since $\|M\| \leq 1$. Thus $\Sigma^{1/2} M \Sigma^{1/2} \in (1+\rho)\mathcal{W}_k$, as claimed. □

Let w^*, A^* be the output of our algorithm. We first claim that the value of the optimal solution is quite small:

Lemma 3.7.5.

$$\left\| \sum_{i=1}^n w_i^* (X_i X_i^\top - I) - \rho A^* \right\|_{\mathcal{W}_{2k}}^* \leq \eta(1+\rho) \ .$$

Proof. Indeed, if we let w be the uniform set of weights over the good points, and we let $A = v v^\top$, then by (3.17), we have

$$\sum_{i=1}^n w_i X_i X_i^\top = \Sigma^{1/2}(I + N)\Sigma^{1/2} \ ,$$

where $\|N\|_{\mathcal{X}_k}^* \leq \eta$, and $\Sigma = I + \rho vv^\top$. Thus we have that

$$\begin{aligned}\left\| \sum_{i=1}^n w_i (X_i X_i^\top - I) - \rho vv^\top \right\|_{\mathcal{W}_{2k}}^* &= \|\Sigma^{1/2} N \Sigma^{1/2}\|_{\mathcal{W}_{2k}}^* \\ &= \max_{M \in \mathcal{W}_k} \left| \mathrm{tr}(\Sigma^{1/2} N \Sigma^{1/2} M) \right| \\ &= \max_{M \in \mathcal{W}_k} \left| \mathrm{tr}(N \Sigma^{1/2} M \Sigma^{1/2}) \right| \\ &\leq (1+\rho) \|N\|_{\mathcal{W}_{2k}}^* \ ,\end{aligned}$$

by Lemma 3.7.4. □

We now show that this implies the following:

Lemma 3.7.6. *$v^\top A^* v \geq 1 - (2+3\rho)\eta/\rho$.*

Proof. By (3.17), we know that we may write $\sum_{i=1}^n w_i (X_i X_i^\top - I) = w^g \rho vv^\top + B - (1-w^g)I + N$, where $B = \sum_{i \in S_{\mathrm{bad}}} w_i X_i X_i^\top$, and $\|N\|_{\mathcal{W}_k}^* \leq (1+\rho)\eta$. Thus, by Lemma 3.7.5 and the triangle inequality, we have that

$$\begin{aligned}\left\| w^g \rho vv^\top + B - \rho A \right\|_{\mathcal{W}_k}^* &\leq \eta + \|N\|_{\mathcal{W}_k}^* + (1-w^g)\|I\|_{\mathcal{W}_k}^* + (1-w^g)\|\rho A\|_{\mathcal{W}_k}^* \\ &\leq (1+\rho)\eta + \varepsilon + \rho\varepsilon \\ &\leq (1+2\rho)\eta + \varepsilon \ .\end{aligned}$$

Now, since $vv^\top \in \mathcal{W}_k$, the above implies that

$$|w^g \rho + v^\top B v - \rho v^\top A^* v| \leq (1+2\rho)\eta + \varepsilon \ ,$$

which by a further triangle inequality implies that

$$|\rho(1 - v^\top A^* v) + v^\top B v| \leq (1+2\rho)\eta + \varepsilon + \varepsilon\rho \leq (2+3\rho)\eta \ .$$

Since $0 \leq v^\top A^* v \leq 1$ (since $A \in \mathcal{X}_k$) and B is PSD, this implies that in fact, we have

$$0 \leq \rho(1 - v^\top A^* v) \leq (2+3\rho)\eta \ .$$

Hence $v^\top A^* v \geq 1 - (2+3\rho)\eta/\rho$, as claimed. □

Let $\gamma = (2+3\rho)\eta/\rho$. The lemma implies that the top eigenvalue of A^* is at least $1-\gamma$. Moreover, since $A^* \in \mathcal{X}_k$, as long as $\gamma \leq 1/2$, this implies that the top eigenvector of A^* is unique up to sign. By the constraint that $\eta \leq O(\min(\rho, 1))$, for an appropriate choice of constants, we that $\gamma \leq 1/10$, and so this condition is satisfied. Recall that u is the top eigenvector of A^*. Since $\mathrm{tr}(A^*) = 1$ and A^* is PSD, we may write $A^* = \lambda_1 u u^\top + A_1$, where u is the top eigenvector of A^*, $\lambda_1 \geq 1-\gamma$, and $\|A_1\| \leq \gamma$. Thus, by the triangle inequality, this implies that

$$\|\rho(vv^\top - \lambda_1 uu^\top) + B\|^*_{\mathcal{X}_{2k}} \leq O(\rho\gamma)$$

which by a further triangle inequality implies that

$$\|\rho(vv^\top - uu^\top) + B\|^*_{\mathcal{X}_{2k}} \leq O(\rho\gamma) \ . \tag{3.18}$$

We now show this implies the following intermediate result:

Lemma 3.7.7. *$(v^\top u)^2 \geq 1 - O(\gamma)$.*

Proof. By Lemma 3.7.6, we have that $v^\top A^* v = \lambda_1 (v^\top u)^2 + v^\top A_1 v \geq 1 - \gamma$. In particular, this implies that $(v^\top u)^2 \geq (1-2\gamma)/\lambda_1 \geq 1 - 3\gamma$, since $1 - \gamma \leq \lambda \leq 1$. □

We now wish to control the spectrum of B. For any subsets $S, T \subseteq [d]$, and for any vector x and any matrix M, let x_S denote x restricted to S and $M_{S,T}$ denote the matrix restricted to the rows in S and the columns in T. Let I be the support of u, and let J be the support of the largest k elements of v.

Lemma 3.7.8. *$\|B_{I,I}\|_2 \leq O(\rho\gamma)$.*

Proof. Observe that the condition (3.18) immediately implies that

$$\|\rho(v_I v_I^\top - u_I u_I^\top) + B_{I,I}\|_2 \leq c\rho\gamma \ , \tag{3.19}$$

for some c, since any unit vector x supported on I satisfies $xx^\top \in \mathcal{X}_{2k}$. Suppose that $\|B_{I,I}\| \geq C\gamma$ for some sufficiently large C. Then (3.19) immediately implies that $\|\rho(v_I v_I^\top - u_I u_I^\top)\|_2 \geq (C-c)\rho\gamma$. Since $(v_I v_I^\top - u_I u_I^\top)$ is clearly rank 2, and satisfies $\mathrm{tr}(v_I v_I^\top - u_I u_I^\top) = 1 - \|u_I\|_2^2 \geq 0$, this implies that the largest eigenvalue of $v_I v_I^\top - u_I u_I^\top$ is positive. Let x be the top eigenvector of $v_I v_I^\top - u_I u_I^\top$. Then, we have $x^\top(v_I v_I^\top - u_I u_I^\top)x + x^\top Bx = (C-c)\rho\gamma + x^\top Bx \geq (C-c)\rho\gamma$ by the PSD-ness of B. If $C > c$, this contradicts (3.19), which proves the theorem. □

This implies the following corollary:

Corollary 3.7.9. *$\|u_I\|_2^2 \geq 1 - O(\gamma)$.*

Proof. Lemma 3.7.8 and (3.19) together imply that $\|v_I v_I^\top - u_I u_I^\top\|_2 \leq O(\gamma)$. The desired bound then follows from a reverse triangle inequality. □

We now show this implies a bound on $B_{J\setminus I, J\setminus I}$:

Lemma 3.7.10. *$\|B_{J\setminus I, J\setminus I}\|_2 \leq O(\rho\gamma)$.*

Proof. Suppose $\|B_{J\setminus I, J\setminus I}\| \geq C\gamma$ for some sufficiently large C. Since u is zero on $J\setminus I$, (3.18) implies that

$$\|\rho v_{J\setminus I} v_{J\setminus I}^\top + B_{J\setminus I, J\setminus I}\|_2 \leq c\rho\gamma \ ,$$

for some universal c. By a triangle inequality, this implies that $\|v_{J\setminus I}\|_2^2 = \|v_{J\setminus I} v_{J\setminus I}^\top\|_2 \geq (C-c)\gamma$. Since v is a unit vector, this implies that $\|v_I\|_2^2 \leq 1 - (C-c)\gamma$, which for a sufficiently large C, contradicts Corollary 3.7.9. □

We now invoke the following general fact about PSD matrices:

Lemma 3.7.11. *Suppose M is a PSD matrix, written in block form as*

$$M = \begin{pmatrix} C & D \\ D^\top & E \end{pmatrix} .$$

Suppose furthermore that $\|C\|_2 \leq \xi$ and $\|E\|_2 \leq \xi$. Then $\|M\|_2 \leq O(\xi)$.

Proof. It is easy to see that $\|M\|_2 \leq O(\max(\|C\|_2, \|D\|_2, \|E\|_2))$. Thus it suffices to bound the largest singular value of D. For any vectors ϕ, ψ with appropriate dimension, we have that

$$(\phi^\top \ \ -\psi^\top)\ M \begin{pmatrix} \phi \\ -\psi \end{pmatrix} = \phi^\top A\phi - 2\phi^\top D\psi + \psi^\top C\psi \geq 0 \ ,$$

which immediately implies that the largest singular value of D is at most $(\|A\|_2 + \|B\|_2)/2$, which implies the claim. □

Therefore, Lemmas 3.7.8 and 3.7.10 together imply:

Corollary 3.7.12. $\|v_{I\cup J} v_{I\cup J}^\top - u_{I\cup J} u_{I\cup J}^\top\|_2 \leq O(\gamma)$.

Proof. Observe (3.18) immediately implies that $\|\rho(v_{I\cup J} v_{I\cup J}^\top - u_{I\cup J} u_{I\cup J}^\top) + B_{I\cup J, I\cup J}\|_2 \leq O(\rho\gamma)$, since $|I \cup J| \leq 2k$. Moreover, Lemmas 3.7.8 and 3.7.10 with Lemma 3.7.11 imply that $\|B_{I\cup J, I\cup J}\|_2 \leq O(\rho\gamma)$, which immediately implies the statement by a triangle inequality. □

Finally, we show this implies $\|vv^\top - u_J u_J^\top\|_2 \leq O(\gamma)$, which is equivalent to the theorem.

Proof of Theorem 3.7.3. We will in fact show the slightly stronger statement, that $\|uu^\top - v_J v_J^\top\|_F \leq O(\gamma)$. Observe that since $uu^\top - vv^\top$ is rank 2, Corollary 3.7.12 implies that $\|v_{I\cup J} v_{I\cup J}^\top - u_{I\cup J} u_{I\cup J}^\top\|_F \leq O(\gamma)$, since for rank two matrices, the spectral and Frobenius norm are off by a constant factor. We have

$$\|uu^\top - vv^\top\|_F^2 = \sum_{(i,j)\in I\cap J \times I\cap J} (u_i u_j - v_i v_j)^2 + \sum_{(i,j)\in I\times I \setminus J\times J} (v_i v_j)^2 + \sum_{(i,j)\in J\times J \setminus I\times I} (u_i u_j)^2 \ .$$

We haveÂă

$$\sum_{(i,j)\in I\cap J \times I\cap J} (u_i u_j - v_i v_j)^2 + \sum_{(i,j)\in J\times J \setminus I\times I} (u_i u_j)^2 \leq \|v_{I\cup J} v_{I\cup J}^\top - u_{I\cup J} u_{I\cup J}^\top\|^2 \leq O(\gamma) \ ,$$

by Corollary 3.7.12. Moreover, we have that

$$\begin{aligned}\sum_{(i,j)\in I\times I\setminus J\times J}(v_iv_j)^2 &\leq 2\left(\sum_{(i,j)\in I\times I\setminus J\times J}(v_iv_j-u_iu_j)^2+\sum_{(i,j)\in I\times I\setminus J\times J}(u_iu_j)^2\right)\\ &\leq 2\left(\|v_{I\cup J}v_{I\cup J}^\top-u_{I\cup J}u_{I\cup J}^\top\|^2+\sum_{(i,j)\in I\times I\setminus J\times J}(u_iu_j)^2\right)\\ &\leq 2\left(\|v_{I\cup J}v_{I\cup J}^\top-u_{I\cup J}u_{I\cup J}^\top\|^2+\sum_{(i,j)\in J\times J\setminus I\times I}(u_iu_j)^2\right)\\ &\leq O(\gamma)\ .\end{aligned}$$

since $J \times J$ contains the k^2 largest entries of $uu^\top$. This completes the proof. □

Chapter 4

Convex Programming III: Sum of Squares and Clustering Mixture Models

Someday, the cold rain will become
warm tears and wash away.
It's alright. This downpour
is just a passing storm.

In this section, we will explore connections between the ideas we've been developing in this thesis, and a number of other problems in high dimensional statistical estimation. In particular, we give new algorithms for the following problems.

1. **Learning Δ-separated mixture models:** Given n samples $X_1, \ldots, X_n \in \mathbb{R}^d$ from a mixture of k probability distributions $D_1, \ldots, D_k$ on $\mathbb{R}^d$ with means $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ and covariances $\Sigma_1, \ldots, \Sigma_k \preceq I$, where $\|\mu_i - \mu_j\|_2 \geq \Delta$, estimate $\mu_1, \ldots, \mu_k$.[1]

2. **Robust mean estimation:** Perhaps our favorite problem at this point: given

[1] A mixture model consists of probability distributions $D_1, \ldots, D_k$ on $\mathbb{R}^d$ and mixing weights $\lambda_1, \ldots, \lambda_k \geq 0$ with $\sum_{i \leq k} \lambda_i = 1$. The distribution D_i has mean μ_i. Each sample x_j is generated by first sampling a component $i \in [k]$ according to the weights λ, then sampling $x_j \sim D_i$.

n vectors $X_1, \ldots, X_n \in \mathbb{R}^d$, of which a $(1-\varepsilon)$-fraction are samples from a probability distribution D with mean μ and covariance $\Sigma \preceq I$ and the remaining ε-fraction are arbitrary vectors (which may depend on the $(1-\varepsilon)n$ samples from D), estimate μ.

Mixture models, and especially Gaussian mixture models (where $D_1, \ldots, D_k$ are Gaussian distributions) have been studied since Pearson in 1894 [Pea94]. Work in theoretical computer science dates at least to the pioneering algorithm of Dasgupta in 1999 [Das99], which has been followed by numerous other algorithms and lower bounds [Wu83, DS07, AK05, VW02, KK10, AM05, FSO06, KMV10, BS10b, MV10, HK13, ABG$^+$14, BCMV14, DK14, SOAJ14, HP15b, XHM16, GHK15, LS17, RV17, DTZ17].

Though outwardly rather different, mixture model learning and robust estimation share some underlying structure. An algorithm for either must identify or otherwise recover information about one or several *structured* subsets of a number of samples $X_1, \ldots, X_n \in \mathbb{R}^d$. In the mixture model case, each collection of all the samples from each distribution D_i is a structured subset. In the robust estimation case there is just one structured subset: the $(1-\varepsilon)n$ samples drawn from the distribution D.[2] Our algorithms are based on new techniques for identifying such structured subsets of points in large data sets.

For mixture models, a special case of our main result yields the first progress in more than 15 years on efficiently clustering mixtures of separated spherical Gaussians. The question here is: if $D_1, \ldots, D_k$ are all Gaussian with covariance identity, what is the minimum cluster separation Δ which allows for a polynomial-time algorithm to estimate $\mu_1, \ldots, \mu_k$ from poly(k, d) samples from the mixture model? When $k = d$, the guarantees of the previous best algorithms for this problem, which require $\Delta \geq O(k^{1/4})$, are captured by a simple greedy clustering algorithm, sometimes called *single-linkage clustering*: when $\Delta \geq O(k^{1/4})$, with high probability every pair of samples from the same cluster is closer in Euclidean distance than every pair of samples from differing clusters.

[2] The recent work [CSV17] codifies this similarity by unifying both these problems into what they call a list-decodable learning setting.

We break this single-linkage clustering barrier: for every $\gamma > 0$ we give a $\mathrm{poly}(k, d)$*-time algorithm for this problem when $\Delta > k^\gamma$.* Our results extend to any k and d. In this more general setting the previous-best algorithms combine spectral dimension reduction (by projecting the samples to the top eigenvectors of an empirical covariance matrix) with single-linkage clustering [VW02]. These algorithms require separation $\Delta \geq O(\min(d, k)^{1/4})$, while our algorithms continue to tolerate separation $\Delta > k^\gamma$ for any $\gamma > 0$.[3]

Our algorithm relies on novel use of higher moments (in fact, $O(1/\gamma)$ moments) of the underlying distributions D_i. Our main technical contribution is a new algorithmic technique for finding either a structured subset of data points or the empirical mean of such a subset when the subset consists of independent samples from a distribution D which has bounded higher-order moments *and there is a simple certificate of this boundedness.* This technique leverages the Sum of Squares (SoS) hierarchy of semidefinite programs (SDPs), and in particular a powerful approach for designing SoS-based algorithms in machine learning settings, developed and used in [BKS14, BKS15, GM15, BM16, HSS15, MSS16, PS17].

This SoS approach to unsupervised learning rests on a notion of *simple identifiability proofs:* the main step in designing an algorithm using SoS to recover some parameters θ from samples $x_1, \ldots, x_n \sim p(x \mid \theta)$ is to prove in a restricted proof system that θ is likely to be uniquely identifiable from $x_1, \ldots, x_n$. We develop this thoroughly later on, but roughly speaking one may think of this as requiring the identifiability proof to use only simple inequalities, such as Cauchy-Schwarz and Hölder's inequality, applied to low-degree polynomials. The simple identifiability proofs we construct for both the mixture models and robust estimation settings are heavily inspired by the robust estimation algorithms studied throughout this thesis.

[3] In the years since an algorithm obtaining $\Delta \geq O(\min(d, k)^{1/4})$ was achieved by [VW02] there has been progress in extending similar results for more general clustering settings. In fact, the algorithm of [VW02] already tolerates isotropic, log-concave distributions, and allows for each component to have a distinct variance $\sigma_i^2 \in \mathbb{R}$, with the separation condition becoming $\|\mu_i - \mu_j\|_2 > \max(\sigma_i, \sigma_j) \min(d, k)^{1/4}$. Later works such as [AM05, KK10, AS12] continued to generalize these results to broader clustering settings. Most related to the present work are spectral algorithms which weaken log-concavity to a bounded-covariance assumption, at the cost of requiring separation $\Delta > \sqrt{k}$.

4.1 Results

Both of the problems we study have a long history; for now we just note some highlights and state our main results.

Mixture models The problem of learning mixture models dates to Pearson in 1894, who invented the method of moments in order to separate a mixture of two Gaussians [Pea94]. Mixture models have since become ubiquitous in data analysis across many disciplines [TSM85, MP04]. In recent years, computer scientists have devised many ingenious algorithms for learning mixture models as it became clear that classical statistical methods (e.g. maximum likelihood estimation) often suffer from computational intractability, especially when there are many mixture components or the components are high dimensional.

A highlight of this work is a series of algorithmic results when the components of the mixture model are Gaussian [Das99, DS07, AK05, VW02]. Here the main question is: how small can the cluster separation Δ be such that there exists an algorithm to estimate $\mu_1, \ldots, \mu_k$ from samples $x_1, \ldots, x_n$ in $\text{poly}(k, d)$ time (hence also using $n = \text{poly}(k, d)$ samples)? Focusing for simplicity on spherical Gaussian components (i.e. with covariance equal to the identity matrix I) and with number of components similar to the ambient dimension of the data (i.e. $k = d$) and uniform mixing weights (i.e. every cluster has roughly the same representation among the samples), the best result in previous work gives a $\text{poly}(k)$-time algorithm when $\Delta \geq k^{1/4}$.

Separation $\Delta = k^{1/4}$ represents a natural algorithmic barrier: when $\Delta \geq k^{1/4}$, *every pair of samples from the same cluster are closer to each other in Euclidean distance than are every pair of samples from distinct clusters (with high probability)*, while this is no longer true if $\Delta < k^{1/4}$. Thus, when $\Delta \geq k^{1/4}$, a simple greedy algorithm correctly clusters the samples into their components (this algorithm is sometimes called *single-linkage clustering*). On the other hand, standard information-theoretic arguments show that the means remain approximately identifiable from $\text{poly}(k, d)$ samples when Δ is as small as $O(\sqrt{\log k})$, but these methods yield only exponential-time algorithms.[4] Nonetheless, despite substantial attention, this $\Delta = k^{1/4}$ barrier representing the breakdown of single-linkage clustering has stood for nearly 20 years.

We prove the following main theorem, breaking the single-linkage clustering barrier.

Theorem 4.1.1 (Informal, special case for uniform mixture of spherical Gaussians)**.** *For every $\gamma > 0$ there is an algorithm with running time $(dk)^{O(1/\gamma^2)}$ using at most $n \leq k^{O(1)} d^{O(1/\gamma)}$ samples which, given samples $x_1, \ldots, x_n$ from a uniform mixture of k spherical Gaussians $\mathcal{N}(\mu_i, I)$ in d dimensions with means $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ satisfying $\|\mu_i - \mu_j\|_2 \geq k^\gamma$ for each $i \neq j$, returns estimators $\hat{\mu}_1, \ldots, \hat{\mu}_k \in \mathbb{R}^d$ such that $\|\hat{\mu}_i - \mu_i\|_2 \leq 1/\operatorname{poly}(k)$ (with high probability).*

We pause here to make several remarks about this theorem. Our algorithm makes novel use of higher order moments of Gaussian (and sub-Gaussian) distributions. Most previous work for efficiently learning well-separated mixtures either used only second-order moment information, and required separation $\Delta \geq \Omega(\sqrt{k})$, or made mild use of log-concavity to improve this to $k^{1/4}$, whereas we use $O(1/\gamma)$ moments.

The guarantees of our theorem hold well beyond the Gaussian setting; the theorem applies to any mixture model with k^γ separation and whose component distributions $D_1, \ldots, D_k$ are what we term $O(1/\gamma)$-*explicitly bounded.* We define this notion formally below, but roughly speaking, a t-explicitly bounded distribution D has t-th moments obeying a subgaussian-type bound—that is, for every unit vector $u \in \mathbb{R}^d$ one has $\mathbb{E}_{Y \sim D} |\langle Y, u \rangle|^t \leq t^{t/2}$—and there is a certain kind of *simple certificate* of this fact, namely a low-degree Sum of Squares proof. Among other things, this means the theorem also applies to mixtures of symmetric product distributions with bounded moments.

For mixtures of distributions with sufficiently-many bounded moments (such as Gaussians), our guarantees go even further. We show that using $d^{O(\log k)^2}$ time and

[4] Recent and sophisticated arguments show that the means are identifiable (albeit inefficiently) with error depending only on the number of samples and not on the separation Δ even when $\Delta = O(\sqrt{\log k})$ [RV17].

$d^{O(\log k)}$ samples, we can recover the means to error $1/\operatorname{poly}(k)$ even if the separation is only $C\sqrt{\log k}$ for some universal constant C. Strikingly, [RV17] show that any algorithm that can learn the means nontrivially given separation $o(\sqrt{\log k})$ must require super-polynomial samples and time. Our results show that just above this threshold, it is possible to learn with just quasipolynomially many samples and time.

Finally, throughout the paper we state error guarantees roughly in terms of obtaining $\hat{\mu_i}$ with $\|\hat{\mu}_i - \mu_i\|_2 \leq 1/\operatorname{poly}(k) \ll k^\gamma$, meaning that we get ℓ_2 error which is much less than the true separation. In the special case of spherical Gaussians, we note that we can use our algorithm as a warm-start to recent algorithms due to [RV17], and achieve error δ using $\operatorname{poly}(m, k, 1/\delta)$ additional runtime and samples for some polynomial independent of γ.

Robust mean estimation While previously in this thesis, we are able to give essentially tight results for mean estimation when the distribution is Gaussian, or subgaussian with isotropic covariance, the state of affairs for general sub-Gaussian distributions is somewhat worse. For general sub-Gaussian distributions with unknown variance $\Sigma \preceq I$, the best known efficient algorithms achieve only $O(\varepsilon^{1/2})$ error (see Chapter 5, also [SCV18]). We substantially improve this, under a slightly stronger condition than sub-Gaussianity. Recall that a distribution D with mean μ over $\mathbb{R}^d$ is sub-Gaussian if for every unit vector u and every $t \in \mathbb{N}$ even, the following moment bound holds:

$$\mathop{\mathbb{E}}_{X \sim D} \langle u, X - \mu \rangle^t \leq t^{t/2} \; .$$

Informally stated, our algorithms will work under the condition that this moment bound can be certified by a low degree SoS proof, for all $s \leq t$. We call such distributions *t-explicitly bounded* (we are ignoring some parameters, see Definition 4.3.1 for a formal definition). This class captures many natural sub-Gaussian distributions, such as Gaussians, product distributions of sub-Gaussians, and rotations thereof (see Appendix D.1.1). For such distributions, we show:

Theorem 4.1.2 (informal, see Theorem 4.6.1)**.** *Fix $\varepsilon > 0$ sufficiently small and let $t \geq 4$. Let D be a $O(t)$-explicitly bounded distribution over $\mathbb{R}^d$ with mean μ^*. There is an algorithm with sample complexity $d^{O(t)}(1/\varepsilon)^{O(1)}$ running time $(d^t\varepsilon)^{O(t)}$ such that given an ε-corrupted set of samples of sufficiently large size from D, outputs μ so that with high probability $\|\mu - \mu^*\|_2 \leq O(\varepsilon^{1-1/t})$.*

As with mixture models, we can push our statistical rates further, if we are willing to tolerate quasipolynomial runtime and sample complexity. In particular, we can obtain error $O(\varepsilon\sqrt{\log 1/\varepsilon})$ with $d^{O(\log 1/\varepsilon)}$ samples and $d^{O(\log 1/\varepsilon)^2}$ time.

4.1.1 Related work

Mixture models The literature on mixture models is vast so we cannot attempt a full survey here. The most directly related line of work to our results studies mixtures models under mean-separation conditions, and especially mixtures of Gaussians, where the number k of components of the mixture grows with the dimension d [Das99, DS07, AK05, VW02]. The culmination of these works is the algorithm of Vempala and Wang, which used spectral dimension reduction to improve on the $d^{1/4}$ separation required by previous works to $k^{1/4}$ in ℓ_2 distance for $k \leq d$ spherical Gaussians in d dimensions. Concretely, they show the following:

Theorem 4.1.3 ([VW02], informal)**.** *There is a constant $C > 0$ and an algorithm with running time* $\mathrm{poly}(k, d)$ *such that for every $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ and $\sigma_1, \ldots, \sigma_k > 0$, satisfying*

$$\|\mu_i - \mu_j\|_2 > C \max(\sigma_i, \sigma_j) k^{1/4} \log^{1/4}(d)$$

with high probability the algorithm produces estimates $\hat{\mu_1}, \ldots, \hat{\mu_k}$ with $\|\mu_i - \hat{\mu_i}\|_2 \leq 1/\mathrm{poly}(k)$, given $\mathrm{poly}(k, d)$ *samples from a mixture $\frac{1}{k}\sum_{i \leq k} \mathcal{N}(\mu_i, \sigma_i I)$.*

The theorem extends naturally to isotropic log-concave distributions; our main theorem generalizes to distributions with explicitly bounded moments. These families of distributions are not strictly comparable.

Other works have relaxed the requirement that the underlying distributions be Gaussian [KK10, AM05]; to second-moment moment boundedness instead of to log-concavity; these algorithms typically tolerate separation of order $\sqrt{k}$ rather than $k^{1/4}$. Our work can be thought of as a generalization of these algorithms to use boundedness of higher moments. One recent work in this spirit uses SDPs to cluster mixture models under separation assumptions [MVW17]; the authors show that a standard SDP relaxation of k-means achieves guarantees comparable to previously-known specially-tailored mixture model algorithms.

Information-theoretic sample complexity: Recent work of [RV17] considers the Gaussian mixtures problem in an information-theoretic setting: they show that there is some constant C so that if the means are pairwise separated by at least $C\sqrt{\log k}$, then the means can be recovered to arbitrary accuracy (given enough samples). They give an efficient algorithm which, warm-started with sufficiently-good estimates of the means, improves the accuracy to δ using $\mathrm{poly}(1/\delta, d, k)$ additional samples. However, their algorithm for providing this warm start requires time exponential in the dimension d. Our algorithm requires somewhat larger separation but runs in polynomial time. Thus by combining the techniques in the spherical Gaussian setting we can estimate the means with ℓ_2 error δ in polynomial time using an extra $\mathrm{poly}(1/\delta, d, k)$ samples, when the separation is at least k^γ, for any $\gamma > 0$.

Fixed number of Gaussians in many dimensions: Other works address parameter estimation for mixtures of $k \ll d$ Gaussians (generally $k = O(1)$ and d grows) under weak identifiability assumptions [KMV10, BS10b, MV10, HP15b]. In these works the only assumptions are that the component Gaussians are statistically distinguishable; the goal is to recover their parameters of the underlying Gaussians. It was shown in [MV10] that algorithms in this setting provably require $\exp(k)$ samples and running time. The question addressed in our paper is whether this lower bound is avoidable under stronger identifiability assumptions. A related line of work addresses proper learning of mixtures of Gaussians [FSO06, DK14, SOAJ14, LS17], where the goal is to output a mixture of Gaussians which is close to the unknown mixture in total-variation distance, avoiding the $\exp(k)$ parameter-learning sample-complexity lower bound. These algorithms achieve $\mathrm{poly}(k, d)$ sample complexity, but they all require $\exp(k)$ running time, and moreover, do not provide any guarantee that the parameters of the distributions output are close to those for the true mixture.

Tensor-decomposition methods: Another line of algorithms focus on settings where the means satisfy algebraic non-degeneracy conditions, which is the case for instance in smoothed analysis settings [HK13, ABG$^+$14, GHK15]. These algorithms are typically based on finding a rank-one decomposition of the empirical 3rd or 4th moment tensor of the mixture; they heavily use the special structure of these moments for Gaussian mixtures. One paper we highlight is [BCMV14], which also uses much higher moments of the distribution. They show that in the smoothed analysis setting, the ℓth moment tensor of the distribution has algebraic structure which can be algorithmically exploited to recover the means. Their main structural result holds only in the smoothed analysis setting, where samples from a mixture model on perturbed means are available.

In contrast, we do not assume any non-degeneracy conditions and use moment information only about the individual components rather than the full mixture, which always hold under separation conditions. Moreover, our algorithms do not need to know the exact structure of the 3rd or 4th moments. In general, clustering-based algorithms like ours seem more robust to modelling errors than algebraic or tensor-decomposition methods.

Expectation-maximization (EM): EM is the most popular algorithm for Gaussian mixtures in practice, but it is notoriously difficult to analyze theoretically. The works [DS07, BWY14, DTZ17, XHM16] offer some theoretical guarantees for EM, but non-convergence results are a barrier to strong theoretical guarantees [Wu83].

SoS algorithms for unsupervised learning SoS algorithms for unsupervised learning obtain the best known polynomial-time guarantees for many problems, including dictionary learning, tensor completion, and others [BKS14, BKS15, GM15, HSS15, MSS16, BM16, PS17]. While the running times of such algorithms are often large polynomials, due to the need to solve large SDPs, insights from the SoS algorithms have often been used in later works obtaining fast polynomial running times [HSSS16, SS17, HKP+17]. This lends hope that in light of our results there is a practical algorithm to learn mixture models under separation $k^{1/4-\varepsilon}$ for some $\varepsilon > 0$.

4.1.2 Organization

In Section 4.2 we discuss at a high level the ideas in our algorithms and SoS proofs. In Section 4.3 we give standard background on SoS proofs. Section 4.4 discusses the important properties of the family of polynomial inequalities we use in both algorithms. Section 4.5 and Section 4.6 state our algorithms formally and analyze them. Finally, Section 4.7 describes the polynomial inequalities our algorithms employ in more detail.

4.2 Techniques

In this section we give a high-level overview of the main ideas in our algorithms. First, we describe the proofs-to-algorithms methodology developed in recent work on SoS algorithms for unsupervised learning problems. Then we describe the core of our algorithms for mixture models and robust estimation: a simple proof of identifiability of the mean of a distribution D on $\mathbb{R}^d$ from samples $X_1, \ldots, X_n$ when some fraction of the samples may not be from D at all.

4.2.1 Proofs to algorithms for machine learning: the SoS method

The Sum of Squares (SoS) hierarchy is a powerful tool in optimization, originally designed to approximately solve systems of polynomial equations via a hierarchy of increasingly strong but increasingly large semidefinite programming (SDP) relaxations (see [BS14] and the references therein). There has been much recent interest in using the SoS method to solve unsupervised learning problems in generative models [BKS14, BKS15, GM15, HSS15, MSS16, PS17]. .

By now there is an established method for designing such SoS-based algorithms, which we employ in this paper. Consider a generic statistical estimation setting: there is a vector $\theta^* \in \mathbb{R}^k$ of parameters, and given some samples $x_1, \ldots, x_n \in \mathbb{R}^d$ sampled iid according to $p(x \,|\, \theta^*)$, one wants to recover some $\hat{\theta}(x_1, \ldots, x_n)$ such that $\|\theta^* - \hat{\theta}\| \leq \delta$ (for some appropriate norm $\| \cdot \|$ and $\delta \geq 0$). One says that θ^* is *identifiable* from $x_1, \ldots, x_n$ if, for any θ with $\|\theta^* - \theta\| > \delta$, one has $\Pr(x_1, \ldots, x_n \,|\, \theta) \ll \Pr(x_1, \ldots, x_n \,|\, \theta^*)$. Often mathematical arguments for identifiability proceed via concentration of measure arguments culminating in a union bound over every possible θ with $\|\theta^* - \theta\| > \delta$. Though this would imply θ could be recovered via brute-force search, this type of argument generally has no implications for efficient algorithms.

The SoS proofs-to-algorithms method prescribes designing a simple proof of identifiability of θ from samples $x_1, \ldots, x_n$. Here "simple" has a formal meaning: the proof should be captured by the low-degree SoS proof system. The SoS proof system can reason about equations and inequalities among low-degree polynomials. Briefly, if $p(y_1, \ldots, y_m)$ and $q(y_1, \ldots, y_m)$ are polynomials with real coefficients, and for every $y \in \mathbb{R}^m$ with $p(y) \geq 0$ it holds also that $q(y) \geq 0$, the SoS proof system can deduce that $p(y) \geq 0$ implies $q(y) \geq 0$ if there is a simple certificate of this implication: polynomials $r(y), s(y)$ which are sums-of-squares, such that $q(y) = r(y) \cdot q(y) + s(y)$. (Then r, s form an SoS proof that $p(y) \geq 0$ implies $q(y) \geq 0$.)

Remarkably, many useful polynomial inequalities have such certificates. For example, the usual proof of the Cauchy-Schwarz inequality $\langle y, z \rangle^2 \leq \|y\|_2^2 \|z\|_2^2$, where y, z are m-dimensional vectors, actually shows that the polynomial $\|y\|_2^2 \|z\|_2^2 - \langle y, z \rangle^2$ is a sum-of-squares in y and z. The simplicity of the certificate is measured by the degree of the polynomials r and s; when these polynomials have small (usually constant) degree there is hope of transforming SoS proofs into polynomial-time algorithms. This transformation is possible because (under mild assumptions on p and q) the set of low-degree SoS proofs is in fact captured by a polynomial-size semidefinite program.

Returning to unsupervised learning, the concentration/union-bound style of identifiability proofs described above are almost never captured by low-degree SoS proofs. Instead, the goal is to design

1. A system of constant-degree polynomial equations and inequalties $\mathcal{A} = \{p_1(\theta) =$

$0, \ldots, p_m(\theta) = 0, q_1(\theta) \geq 0, \ldots, q_m(\theta) \geq 0\}$, where the polynomials p and q depend on the samples $x_1, \ldots, x_n$, such that with high probability θ^* satisfies all the equations and inequalities.

2. A low-degree SoS proof that $\mathcal{A}$ implies $\|\theta - \theta^*\|_2 \leq \delta$ for some small δ and appropriate norm $\| \cdot \|_2$.

Clearly these imply that any solution θ of $\mathcal{A}$ also solves the unsupervised learning problem. It is in general NP-hard to find a solution to a system of low-degree polynomial equations and inequalities.

However, the SoS proof (2) means that such a search can be avoided. Instead, we will relax the set of solutions θ to $\mathcal{A}$ to a simple(er) convex set: the set of *pseudodistributions satisfying* $\mathcal{A}$. We define pseudodistributions formally later, for now saying only that they are the convex duals of SoS proofs which use the axioms $\mathcal{A}$. By this duality, the SoS proof (2) implies not only that any solution θ to $\mathcal{A}$ is a good choice of parameters but also that a good choice of parameters can be extracted from any pseudodistribution satisfying $\mathcal{A}$. (We are glossing over for now that this last step requires some SDP rounding algorithm, since we use only standard rounding algorithms in this paper.)

Thus, the final SoS algorithms from this method take the form: solve an SDP to find a pseudodistribution which satisfies $\mathcal{A}$ and round it to obtain a estimate $\hat{\theta}$ of θ^*. To analyze the algorithm, use the SoS proof (2) to prove that $\|\hat{\theta} - \theta^*\|_2 \leq \delta$.

4.2.2 Hölder's inequality and identifiability from higher moments

Now we discuss the core ideas in our simple SoS identifiability proofs. We have not yet formally defined SoS proofs, so our goal will just be to construct identifiability proofs which are (a) phrased in terms of inequalities of low-degree polynomials and (b) provable using only simple inequalities, like Cauchy-Schwarz and Hölder's inequalities, leaving the formalities for later.

We consider an idealized version of situations we encounter in both the mixture model and robust estimation settings. Let $\mu^* \in \mathbb{R}^d$. Let $X_1, \ldots, X_n \in \mathbb{R}^d$ have the guarantee that for some $T \subseteq [n]$ of size $|T| = \alpha n$, the vectors $\{X_i\}_{i \in T}$ are iid samples from $\mathcal{N}(\mu^*, I)$, a spherical Gaussian centered at μ^*; for the other vectors we make no assumption. The goal is to estimate the mean μ^*.

The system $\mathcal{A}$ of polynomial equations and inequalities we employ will be designed so that a solution to $\mathcal{A}$ corresponds to a subset of samples $S \subseteq [n]$ of size $|S| = |T| = \alpha n$. We accomplish this by identifying S with its 0/1 indicator vector in $\mathbb{R}^n$ (this is standard). The inequalities in $\mathcal{A}$ will enforce the following crucial moment property on solutions: if $\mu = \frac{1}{|S|} \sum_{i \in S} X_i$ is the empirical mean of samples in S and $t \in \mathbb{N}$, then

$$\frac{1}{|S|} \sum_{i \in S} \langle X_i - \mu, u \rangle^t \leq 2 \cdot t^{t/2} \cdot \|u\|_2^t \qquad \text{for all } u \in \mathbb{R}^d \,. \tag{4.1}$$

This inequality says that every one-dimensional projection u of the samples in S, centered around their empirical mean, has a sub-Gaussian empirical t-th moment. (The factor 2 accounts for deviations in the t-th moments of the samples.) By standard concentration of measure, if $\alpha n \gg d^t$ the inequality holds for $S = T$. It turns out that this property can be enforced by polynomials of degree t. (Actually our final construction of $\mathcal{A}$ will need to use inequalities of matrix-valued polynomials but this can be safely ignored here.)

Intuitively, we would like to show that any S which satisfies $\mathcal{A}$ has empirical mean close to μ^* using a low-degree SoS proof,. This is in fact true when $\alpha = 1 - \varepsilon$ for small ε, which is at the core of our robust estimation algorithm. However, in the mixture model setting, when $\alpha = 1/(\# \text{ of components})$, for each component j there is a subset $T_j \subseteq [n]$ of samples from component j which provides a valid solution $S = T_j$ to $\mathcal{A}$. The empirical mean of T_j is close to μ_j and hence not close to μ_i for any $i \neq j$.

We will prove something slightly weaker, which still demonstrates the main idea in our identifiability proof.

Lemma 4.2.1. *With high probability, for every $S \subseteq [n]$ which satisfies* (4.1)*, if $\mu = \frac{1}{|S|} \sum_{i \in S} X_i$ is the empirical mean of samples in S, then $\|\mu - \mu^*\|_2 \leq 4t^{1/2} \cdot (|T|/|S \cap T|)^{1/t}$.*

Notice that a random $S \subseteq [n]$ of size αn will have $|S \cap T| \approx \alpha^2 n$. In this case the lemma would yield the bound $\|\mu - \mu^*\|_2 \leq \frac{4t^{1/2}}{\alpha^{1/t}}$. Thinking of $\alpha \ll 1/t$, this bound improves exponentially as t grows. In the d-dimensional k-component mixture model setting, one has $1/\alpha = \mathrm{poly}(k)$, and thus the bound becomes $\|\mu - \mu^*\|_2 \leq 4t^{1/2} \cdot k^{O(1/t)}$. In a mixture model where components are separated by k^ε, such an estimate is nontrivial when $\|\mu - \mu^*\|_2 \ll k^\varepsilon$, which requires $t = O(1/\varepsilon)$. This is the origin of the quantitative bounds in our mixture model algorithm.

We turn to the proof of Lemma 4.2.1. As we have already emphasized, the crucial point is that this proof will be accomplished using only simple inequalities, avoiding any union bound over all possible subsets S.

Proof of Lemma 4.2.1. Let w_i be the 0/1 indicator of $i \in S$. To start the argument, we expand in terms of samples:

$$\begin{aligned} |S \cap T| \cdot \|\mu - \mu^*\|_2^2 &= \sum_{i \in T} w_i \|\mu - \mu^*\|_2^2 \\ &= \sum_{i \in T} w_i \langle \mu^* - \mu, \mu^* - \mu \rangle \qquad (4.2) \\ &= \sum_{i \in T} w_i \left[\langle X_i - \mu, \mu^* - \mu \rangle + \langle \mu^* - X_i, \mu^* - \mu \rangle \right] . \qquad (4.3) \end{aligned}$$

The key term to bound is the first one; the second amounts to a deviation term. By

Hölder's inequality and for even t,

$$\begin{aligned}\sum_{i\in T} w_i\langle X_i-\mu,\mu^*-\mu\rangle &\le \left(\sum_{i\in T} w_i\right)^{\frac{t-1}{t}}\cdot\left(\sum_{i\in T} w_i\langle X_i-\mu,\mu^*-\mu\rangle^t\right)^{1/t}\\ &\le \left(\sum_{i\in T} w_i\right)^{\frac{t-1}{t}}\cdot\left(\sum_{i\in[n]} w_i\langle X_i-\mu,\mu^*-\mu\rangle^t\right)^{1/t}\\ &\le \left(\sum_{i\in T} w_i\right)^{\frac{t-1}{t}}\cdot 2t^{1/2}\cdot\|\mu^*-\mu\|_2\\ &= |S\cap T|^{\frac{t-1}{t}}\cdot 2t^{1/2}\cdot\|\mu^*-\mu\|_2\,.\end{aligned}$$

The second line follows by adding the samples from $[n]\setminus T$ to the sum; since t is even this only increases its value. The third line uses the moment inequality (4.1). The last line just uses the definition of w.

For the second, deviation term, we use Hölder's inequality again:

$$\sum_{i\in T} w_i\langle\mu^*-X_i,\mu^*-\mu\rangle \le \left(\sum_{i\in T} w_i\right)^{\frac{t-1}{t}}\cdot\left(\sum_{i\in T}\langle\mu^*-X_i,\mu^*-\mu\rangle^t\right)^{1/t}\,.$$

The distribution of μ^*-X_i for $i\in T$ is $\mathcal{N}(0,I)$. By standard matrix concentration, if $|T|=\alpha n\gg d^t$,

$$\sum_{i\in T}\left[(X_i-\mu^*)^{\otimes t/2}\right]\left[(X_i-\mu^*)^{\otimes t/2}\right]^\top \preceq 2|T|\mathop{\mathbb{E}}_{Y\sim\mathcal{N}(0,I)}\left(Y^{\otimes t/2}\right)\left(Y^{\otimes t/2}\right)^\top$$

with high probability and hence, using the quadratic form at $(\mu^*-\mu)^{\otimes t/2}$,

$$\sum_{i\in T}\langle\mu^*-X_i,\mu^*-\mu\rangle^t \le 2|T|t^{t/2}\cdot\|\mu^*-\mu\|_2^t\,.$$

Putting these together and simplifying constants, we have obtained that with high probability,

$$|S\cap T|\cdot\|\mu-\mu^*\|_2^2 \le 4t^{1/2}|T|^{1/t}\cdot|S\cap T|^{(t-1)/t}\cdot\|\mu-\mu^*\|_2$$

which simplifies to

$$|S \cap T|^{1/t} \cdot \|\mu - \mu^*\|_2 \leq 4t^{1/2}|T|^{1/t} . \quad \square$$

4.2.3 From identifiability to algorithms

We now discuss how to use the ideas described above algorithmically for learning well-separated mixture models. The high level idea for robust estimation is similar. Given Lemma 4.2.1, a naive algorithm for learning mixture models would be the following: find a set of points T of size roughly n/k that satisfy the moment bounds described, and simply output their empirical mean. Since by a simple counting argument this set must have nontrivial overlap with the points from some mixture component, Lemma 4.2.1 guarantees that the empirical mean is close to mean of this component.

However, in general finding such a set of points is algorithmically difficult. In fact, it would suffice to find a distribution over such sets of points (since then one could simply sample from this distribution), however, this is just as computationally difficult. The critical insight is that because of the proof of Lemma 4.2.1 only uses facts about low degree polynomials, it suffices to find an object which is indistinguishable from such a distribution, considered as a functional on low-degree polynomials.

The natural object in this setting is a *pseudo-distribution*. Pseudo-distributions form a convex set, and for a set of low-degree polynomial equations and inequalities $\mathcal{A}$, it is possible to find a pseudo-distribution which is indistinguishable from a distribution over solutions to $\mathcal{A}$ (as such a functional) in polynomial time via semidefinite programming (under mild assumptions on $\mathcal{A}$). More specifically, the set of SoS proofs using axioms $\mathcal{A}$ is a semidefinite program (SDP), and the above pseudodistributions form the dual SDP. (We will make these ideas more precise in the next two sections.)

Our algorithm then proceeds via the following general framework: find an appropriate pseudodistribution via convex optimization, then leverage our low-degree sum of squares proofs to show that information about the true clusters can be extracted from this object by a standard SDP rounding procedure.

4.3 Preliminaries

Throughout the paper we let d be the dimensionality of the data, and we will be interested in the regime where d is at least a large constant. We also let $\|v\|_2$ denote the ℓ_2 norm of a vector v, and $\|M\|_F$ to denote the Frobenius norm of a matrix M. We will also give randomized algorithms for our problems that succeed with probability $1-\text{poly}(1/k, 1/d)$; by standard techniques this probability can be boosted to $1-\xi$ by increasing the sample and runtime complexity by a mulitplicative $\log 1/\xi$. Moreover, in accordance with some conventions from the SoS literature, we will often drop the brackets on the outside of expectations.

We now formally define the class of distributions we will consider throughout this paper. At a high level, we will consider distributions which have bounded moments, for which there exists a low degree SoS proof of this moment bound. Formally:

Definition 4.3.1. Let D be a distribution over $\mathbb{R}^d$ with mean μ. For $c \geq 1, t \in \mathbb{N}$, we say that D is t-explicitly bounded with variance proxy σ if for every even $s \leq t$ there is a degree s SoS proof (see Section 4.3.1 for a formal definition) of

$$\vdash_s \; E_{Y \sim D_k} \langle (Y-\mu), u \rangle^s \leq (\sigma s)^{s/2} \|u\|_2^s \, .$$

Equivalently, the polynomial $p(u) = (\sigma s)^{s/2}\|u\|_2^s - E_{Y \sim D_k}\langle (Y-\mu), u\rangle^s$ should be a sum-of-squares. In our typical use case, $\sigma = 1$, we will omit it and call the distribution t-explicitly bounded.

Throughout this paper, since all of our problems are scale invariant, we will assume without loss of generality that $\sigma = 1$. This class of distributions captures a number of natural classes of distributions. Intuitively, if u were truly a vector in $\mathbb{R}^k$ (rather than a vector of indeterminants), then this exactly captures sub-Gaussian type moment. Our requirement is simply that these types of moment bounds not only hold, but also have a SoS proof.

We remark that our results also hold for somewhat more general settings. It is not particularly important that the s-th moment bound has a degree s proof; our techniques can tolerate degree $O(s)$ proofs. Our techniques also generally apply for weaker moment bounds. For instance, our techniques naturally extend to explicitly bounded sub-exponential type distributions in the obvious way. We omit these details for simplicity.

As we show in Appendix D.1.1, this class still captures many interesting types of nice distributions, including Gaussians, product distributions with sub-Gaussian components, and rotations therof. With this definition in mind, we can now formally state the problems we consider in this paper:

Learning well-separated mixture models We first define the class of mixture models for which our algorithm works:

Definition 4.3.2 (t-explicitly bounded mixture model with separation Δ)**.** Let $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ satisfy $\|\mu_i - \mu_j\|_2 > \Delta$ for every $i \neq j$, and let $D_1, \ldots, D_k$ have means $\mu_1, \ldots, \mu_k$, so that each D_i is t-explicitly bounded. Let $\lambda_1, \ldots, \lambda_k \geq 0$ satisfy $\sum_{i \in [k]} \lambda_i = 1$. Together these define a mixture distribution on $\mathbb{R}^d$ by first sampling $i \sim \lambda$, then sampling $x \sim D_i$.

The problem is then:

Problem 4.3.1. Let D be a t-explicitly bounded mixture model in $\mathbb{R}^d$ with separation Δ with k components. Given k, Δ, and n independent samples from D, output $\widehat{\mu}_1, \ldots, \widehat{\mu}_m$ so that with probability at least 0.99, there exists a permutation $\pi : [k] \to [k]$ so that $\|\mu_i - \widehat{\mu}_{\pi(i)}\|_2 \leq \delta$ for all $i = 1, \ldots, k$.

Robust mean estimation We consider the same basic model of corruption as we do throughout this thesis. The problem we consider in this setting is the following:

Problem 4.3.2 (Robust mean estimation)**.** Let D be an $O(t)$-explicitly bounded distribution over $\mathbb{R}^d$ wih mean μ. Given t, ε, and an ε-corrupted set of samples from D, output $\widehat{\mu}$ satisfying $\|\mu - \widehat{\mu}\|_2 \leq O(\varepsilon^{1-1/t})$.

4.3.1 The SoS proof system

We refer the reader to [OZ13, BS14] and the references therein for a thorough exposition of the SoS algorithm and proof system; here we only define what we need.[5]

Let $x_1, \ldots, x_n$ be indeterminates and $\mathcal{A}$ be the set of polynomial equations and inequalities $\{p_1(x) \geq 0, \ldots, p_m(x) \geq 0, q_1(x) = 0, \ldots, q_m(x) = 0\}$. We say that the statement $p(x) \geq 0$ has an SoS proof if there are polynomials $\{r_\alpha\}_{\alpha \subseteq [m]}$ (where α may be a multiset) and $\{s_i\}_{i \in [m]}$ such that

$$p(x) = \sum_{\alpha} r_\alpha(x) \cdot \prod_{i \in \alpha} p_i(x) + \sum_{i \in [m]} s_i(x) q_i(x)$$

and each polynomial $r_\alpha(x)$ is a sum of squares.

If the polynomials $r_\alpha(x) \cdot \prod_{i \in \alpha} p_i(x)$ and $s_i(x)q_i(x)$ have degree at most d, we say the proof has degree at most d, and we write

$$\mathcal{A} \vdash_d p(x) \geq 0 \, .$$

SoS proofs compose well, and we frequently use the following without comment.

Fact 4.3.1. *If $\mathcal{A} \vdash_d p(x) \geq 0$ and $\mathcal{A} \vdash_{d'} q(x) \geq 0$, then $\mathcal{A} \cup \mathcal{B} \vdash_{\max(d,d')} p(x) + q(x) \geq 0$ and $\mathcal{A} \cup \mathcal{B} \vdash_{dd'} p(x)q(x) \geq 0$.*

We turn to the dual objects to SoS proofs. A degree-d pseudoexpectation (for variety we sometimes say "pseudodistribution") is a linear operator $\tilde{\mathbb{E}} : \mathbb{R}[x]_{\leq d} \to \mathbb{R}$, where $\mathbb{R}[x]_{\leq d}$ are the polynomials in indeterminates x with real coefficients, which satisfies the following

1. Normalization: $\tilde{\mathbb{E}}[1] = 1$

2. Positivity: $\tilde{\mathbb{E}}[p(x)^2] \geq 0$ for every p of degree at most $d/2$.

We say that a degree-d pseudoexpectation $\tilde{\mathbb{E}}$ satisfies inequalities and equalities $\{p_1(x) \geq 0, \ldots, p_m(x) \geq 0, q_1(x) = 0, \ldots, q_m(x) = 0\}$ at degree $r \leq d$ if

[5] Our definition of SoS proofs differs slightly from O'Donnell and Zhou's in that we allow proofs to use products of axioms.

1. for every multiset $\alpha \subseteq [m]$ and SoS polynomial $s(x)$ such that the degree of $s(x) \prod_{i \in \alpha} p_i(x)$ is at most r, one has $\tilde{\mathbb{E}}\, s(x) \prod_{i \in \alpha} p_i(x) \geq 0$, and
2. for every $q_i(x)$ and every polynomial $s(x)$ such that the degree of $q_i(x)s(x) \leq r$, one has $\tilde{\mathbb{E}}\, s(x) q_i(x) = 0$.

The main fact relating pseudoexpectations and SoS proofs is:

Fact 4.3.2 (Soundness of SoS proofs, informal)**.** *If $\mathcal{A}$ is a set of equations and inequalities and $\mathcal{A} \vdash_\ell p(x) \geq 0$, and $\tilde{\mathbb{E}}$ is a degree $d > \ell$ pseudodistribution satisfying $\mathcal{A}$ at degree d, then $\tilde{\mathbb{E}}$ satisfies $\mathcal{A} \cup \{p \geq 0\}$ at degree $d - \ell$.*[6]

In Section D.1 we state and prove many basic SoS inequalities that we will require throughout the paper.

Gaussian distributions are explicitly bounded In Section D.1 we show that product distributions (and rotations thereof) with bounded t-th moments are explicitly bounded.

Lemma 4.3.3. *Let D be a distribution over $\mathbb{R}^d$ so that D is a rotation of a product distribution D' where each coordinate X with mean μ of D satisfies*

$$\mathbb{E}[(X - \mu)^s] \leq 2^{-s} \left(\frac{s}{2}\right)^{s/2}$$

Then D is t-explicitly bounded (with variance proxy 1).

(The factors of $\frac{1}{2}$ can be removed for many distributions, including Gaussians.)

4.4 Capturing empirical moments with polynomials

To describe our algorithms we need to describe a system of polynomial equations and inequalities which capture the following problem: among $X_1, \ldots, X_n \in \mathbb{R}^d$, find a subset of $S \subseteq [n]$ of size αn such that the empirical t-th moments obey a moment bound: $\frac{1}{\alpha n} \sum_{i \in S} \langle X_i, u \rangle^t \leq t^{t/2} \|u\|_2^t$ for every $u \in \mathbb{R}^d$.

[6] See [BS17] for a full account of completeness and soundness of SoS.

Let $k, n \in \mathbb{N}$ and let $w = (w_1, \ldots, w_n), \mu = (\mu_1, \ldots, \mu_k)$ be indeterminates. Let

1. $X_1, \ldots, X_n \in \mathbb{R}^d$
2. $\alpha \in [0, 1]$ be a number (the intention is $|S| = \alpha n$).
3. $t \in \mathbb{N}$ be a power of 2, the order of moments to control
4. $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, which will eventually be the means of a k-component mixture model, or when $k = 1$, the true mean of the distribution whose mean we robustly estimate.
5. $\tau > 0$ be some error magnitude accounting for fluctuations in the sizes of clusters (which may be safely ignored at first reading).

Definition 4.4.1. Let $\mathcal{A}$ be the following system of equations and inequalities, depending on all the parameters above.

1. $w_i^2 = w_i$ for all $i \in [n]$ (enforcing that w is a $0/1$ vector, which we interpret as the indicator vector of the set S).
2. $(1 - \tau)\alpha n \leq \sum_{i \in [n]} w_i \leq (1 + \tau)\alpha n$, enforcing that $|S| \approx \alpha n$ (we will always choose $\tau = o(1)$).
3. $\mu \cdot \sum_{i \in [n]} w_i = \sum_{i \in [n]} w_i X_i$, enforcing that μ is the empirical mean of the samples in S
4. $\sum_{i \in [n]} w_i \langle X_i - \mu, \mu - \mu_j \rangle^t \leq 2 \cdot t^{t/2} \sum_{i \in [n]} w_i \|\mu - \mu_j\|_2^t$ for every μ_j among $\mu_1, \ldots, \mu_m$. This enforces that the t-th empirical moment of the samples in S is bounded *in the direction* $\mu - \mu_j$.

Notice that since we will eventually take μ_j's to be unknown parameters we are trying to estimate, the algorithm cannot make use of $\mathcal{A}$ directly, since the last family of inequalities involve the μ_j's. Later in this paper we exhibit a system of inequalities which requires the empirical t-th moments to obey a sub-Gaussian type bound in every direction, hence implying the inequalities here without requiring knowledge of the μ_j's to write down. Formally, we will show:

Lemma 4.4.1. *Let $\alpha \in [0,1]$. Let $t \in \mathbb{N}$ be a power of 2, $t \geq 4$.*[7] *Let $0.1 > \tau > 0$. Let $X_1, \ldots, X_n \in \mathbb{R}^d$. Let D be a $10t$-explicitly bounded distribution.*

There is a family $\widehat{\mathcal{A}}$ of polynomial equations and inequalities of degree $O(t)$ on variables $w = (w_1, \ldots, w_n), \mu = (\mu_1, \ldots, \mu_k)$ and at most $n^{O(t)}$ other variables, whose coefficients depend on $\alpha, t, \tau, X_1, \ldots, X_n$, such that

1. *(Satisfiability) If there $S \subseteq [n]$ of size at least $(\alpha - \tau)n$ so that $\{X_i\}_{i \in S}$ is an iid set of samples from D, and $(1-\tau)\alpha n \geq d^{100t}$, then for d large enough, with probability at least $1 - d^{-8}$, the system $\widehat{\mathcal{A}}$ has a solution over $\mathbb{R}$ which takes w to be the $0/1$ indicator vector of S.*

2. *(Solvability) For every $C \in \mathbb{N}$ there is an $n^{O(Ct)}$-time algorithm which, when $\widehat{\mathcal{A}}$ is satisfiable, returns a degree-Ct pseudodistribution which satisfies $\widehat{\mathcal{A}}$ (up to additive error 2^{-n}).*

3. *(Moment bounds for polynomials of μ) Let $f(\mu)$ be a length-d vector of degree-ℓ polynomials in indeterminates $\mu = (\mu_1, \ldots, \mu_k)$. $\widehat{\mathcal{A}}$ implies the following inequality and the implication has a degree $t\ell$ SoS proof.*

$$\widehat{\mathcal{A}} \vdash_{O(t\ell)} \frac{1}{\alpha n} \sum_{i \in [n]} w_i \langle X_i - \mu, f(\mu) \rangle^t \leq 2 \cdot t^{t/2} \|f(\mu)\|_2^t \,.$$

4. *(Booleanness) $\widehat{\mathcal{A}}$ includes the equations $w_i^2 = w_i$ for all $i \in [n]$.*

5. *(Size) $\widehat{\mathcal{A}}$ includes the inequalities $(1-\tau)\alpha n \leq \sum w_i \leq (1+\tau)\alpha n$.*

6. *(Empirical mean) $\widehat{\mathcal{A}}$ includes the equation $\mu \cdot \sum_{i \in [n]} w_i = \sum_{i \in [n]} w_i X_i$.*

In particular this implies that $\widehat{\mathcal{A}} \vdash_{O(t)} \mathcal{A}$.

The proof of Lemma 4.4.1 can be found in Section 4.7.

Remark 4.4.1 (Numerical accuracy, semidefinite programming, and other monsters). We pause here to address issues of numerical accuracy. Our final algorithms use point

[7] The condition $t \geq 4$ is merely for technical convenience.

2 in Lemma 4.4.1 (itself implemented using semidefinite programming) to obtain a pseudodistribution $\tilde{\mathbb{E}}$ satisfying $\widehat{\mathcal{A}}$ approximately, up to error $\eta = 2^{-n}$ in the following sense: for every r a sum of squares and $f_1, \ldots, f_\ell \in \mathcal{A}$ with $\deg [r \cdot \prod f_i \leq Ct]$, one has $\tilde{\mathbb{E}}\, r \cdot \prod_{i \in \mathcal{A}} f \geq -\eta \cdot \|r\|_2$, where $\|r\|_2$ is ℓ_2 norm of the coefficients of r. Our main analyses of this pseudodistribution employ the implication $\widehat{\mathcal{A}} \vdash \mathcal{B}$ for another family of inequalities $\mathcal{B}$ to conclude that if $\tilde{\mathbb{E}}$ satisfies $\mathcal{A}$ then it satisfies $\mathcal{B}$, then use the latter to analyze our rounding algorithms. Because all of the polynomials eventually involved in the SoS proof $\widehat{\mathcal{A}} \vdash \mathcal{B}$ have coefficients bounded by n^B for some large constant B, it may be inferred that if $\tilde{\mathbb{E}}$ approximately satisfies $\widehat{\mathcal{A}}$ in the sense above, it also approximately satisfies $\mathcal{B}$, with some error $\eta' \leq 2^{-\Omega(n)}$. The latter is a sufficient for all of our rounding algorithms.

Aside from mentioning at a couple key points why our SoS proofs have bounded coefficients, we henceforth ignore all numerical issues. For further discussion of numerical accuracy and well-conditioned-ness issues in SoS, see [O'D17, BS17, RW17]

4.5 Mixture models: algorithm and analysis

In this section we formally describe and analyze our algorithm for mixture models. We prove the following theorem.

Theorem 4.5.1 (Main theorem on mixture models)**.** *For every large-enough $t \in \mathbb{N}$ there is an algorithm with the following guarantees. Let $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, satisfy $\|\mu_i - \mu_j\|_2 \geq \Delta$. Let $D_1, \ldots, D_k$ be $10t$-explicitly bounded, with means $\mu_1, \ldots, \mu_k$. Let $\lambda_1, \ldots, \lambda_k \geq 0$ satisfy $\sum \lambda_i = 1$. Given $n \geq (d^t k)^{O(1)} \cdot (\max_{i \in [m]} 1/\lambda_i)^{O(1)}$ samples from the mixture model given by $\lambda_1, \ldots, \lambda_k, D_1, \ldots, D_k$, the algorithm runs in time $n^{O(t)}$ and with high probability returns $\{\hat{\mu}_1, \ldots, \hat{\mu}_k\}$ (not necessarily in that order) such that*

$$\|\mu_i - \hat{\mu}_i\|_2 \leq \frac{2^{Ct} m^C t^{t/2}}{\Delta^{t-1}}$$

for some universal constant C.

In particular, we note two regimes: if $\Delta = k^\gamma$ for a constant $\gamma > 0$, choosing $t = O(1/\gamma)$ we get that the ℓ_2 error of our estimator is $\mathrm{poly}(1/k)$ for any $O(1/\gamma)$-explicitly bounded distribution, and our estimator requires only $(dk)^{O(1)}$ samples and time. This matches the guarantees of Theorem 4.1.1.

On the other hand, if $\Delta = C'\sqrt{\log k}$ (for some universal C') then taking $t = O(\log k)$ gives error

$$\|\mu_i - \hat{\mu}_i\|_2 \le k^{O(1)} \cdot \left(\frac{\sqrt{t}}{\Delta}\right)^t$$

which, for large-enough C' and t, can be made $1/\mathrm{poly}(k)$. Thus for $\Delta = C'\sqrt{\log k}$ and any $O(\log k)$-explicitly bounded distribuion we obtain error $1/\mathrm{poly}(k)$ with $d^{O(\log k)}$ samples and $d^{O(\log k)^2}$ time.

In this section we describe and analyze our algorithm. To avoid some technical work we analyze the uniform mixtures setting, with $\lambda_i = 1/m$. In Section D.4 we describe how to adapt the algorithm to the nonuniform mixture setting.

4.5.1 Algorithm and main analysis

We formally describe our mixture model algorithm now. We use the following lemma, which we prove in Section 4.5.6. The lemma says that given a matrix which is very close, in Frobenious norm, to the 0/1 indicator matrix of a partition of $[n]$ it is possible to approximately recover the partition. (The proof is standard.)

Lemma 4.5.2 (Second moment rounding, follows from Theorem 4.5.11)**.** *Let $n, m \in \mathbb{N}$ with $m \ll n$. There is a polynomial time algorithm* RoundSecondMoments *with the following guarantees. Suppose $S_1, \ldots, S_m$ partition $[n]$ into m pieces, each of size $\frac{n}{2m} \le |S_i| \le \frac{2n}{m}$. Let $A \in \mathbb{R}^{n \times n}$ be the $0/1$ indicator matrix for the partition S; that is, $A_{ij} = 1$ if $i, j \in S_\ell$ for some ℓ and is 0 otherwise. Let $M \in \mathbb{R}^{n \times n}$ be a matrix with $\|A - M\|_F \le \varepsilon n$. Given M, with probability at least $1 - \varepsilon^2 m^3$ the algorithm returns a partition $C_1, \ldots, C_m$ of $[n]$ such that up to a global permutation of $[m]$, $C_i = T_i \cup B_i$, where $T_i \subseteq S_i$ and $|T_i| \ge |S_i| - \varepsilon^2 m^2 n$ and $|B_i| \le \varepsilon^2 m^2 n$.*

Algorithm 11 Mixture Model Learning

```
function LearnMixtureMeans($t, X_1, \ldots, X_n, \delta, \tau$)
    By semidefinite programming (see Lemma 4.4.1, item 2), find a pseudoexpectation of degree $O(t)$ which satisfies the structured subset polynomials from Lemma 4.4.1, with $\alpha = n/m$ such that $\| \tilde{\mathbb{E}}\, ww^\top \|_F$ is minimized among all such pseudoexpectations.
    Let $M \leftarrow m \cdot \tilde{\mathbb{E}}\, ww^\top$.
    Run the algorithm RoundSecondMoments on $M$ to obtain a partition $C_1, \ldots, C_m$ of $[n]$.
    Run the algorithm EstimateMean from Section 4.6 on each cluster $C_i$, with $\varepsilon = 2^{Ct} t^{t/2} m^4 / \Delta^t$ for some universal constant $C$ to obtain a list of mean estimates $\hat{\mu}_1, \ldots, \hat{\mu}_m$.
    Output $\hat{\mu}_1, \ldots, \hat{\mu}_m$.
```

Remark 4.5.1 (On the use of EstimateMean). As described, LearnMixtureMeans has two phases: a clustering phase and a mean-estimation phase. The clustering phase is the heart of the algorithm; we will show that after running RoundSecondMoments the algorithm has obtained clusters $C_1, \ldots, C_k$ which err from the ground-truth clustering on only a $\frac{2^{O(t)} t^{t/2} \operatorname{poly}(k)}{\Delta^t}$-fraction of points. To obtain estimates $\hat{\mu}_i$ of the underlying means from such a clustering, one simple option is to output the empirical mean of the clusters. However, without additional pruning this risks introducing error in the mean estimates which grows with the ambient dimension d. By using the robust mean estimation algorithm instead to obtain mean estimates from the clusters we obtain errors in the mean estimates which depend only on the number of clusters k, the between-cluster separation Δ, and the number t of bounded moments.

Remark 4.5.2 (Running time). We observe that LearnMixtureMeans can be implemented in time $n^{O(t)}$. The main theorem requires $n \geq k^{O(1)} d^{O(t)}$, which means that the final running time of the algorithm is $(kd^t)^{O(t)}$.[8]

[8] As discussed in Section 4.4, correctness of our algorithm at the level of numerical accuracy requires that the coefficients of every polynomial in the SoS program $\widehat{\mathcal{A}}$ (and every polynomial in the SoS proofs we use to analyze $\widehat{\mathcal{A}}$) are polynomially bounded. This may not be the case if some vectors $\mu_1, \ldots, \mu_m$ have norms $\|\mu_i\|_2 \geq d^{\omega(1)}$. This can be fixed by naively clustering the samples $X_1, \ldots, X_n$ via single-linkage clustering, then running LearnMixtureMeans on each cluster. It is routine to show that the diameter of each cluster output by a naive clustering algorithm is at most $\operatorname{poly}(d, k)$ under our assumptions, and that with high probability single-linkage clustering produces a clustering respecting the distributions D_i. Hence, by centering each cluster before running

4.5.2 Proof of main theorem

In this section we prove our main theorem using the key lemmata; in the following sections we prove the lemmata.

Deterministic Conditions We recall the setup. There are k mean vectors $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, and corresponding distributions $D_1, \ldots, D_k$ where D_j has mean μ_j. The distributions D_j are $10t$-explicitly bounded for a choice of t which is a power of 2. Vectors $X_1, \ldots, X_n \in \mathbb{R}^d$ are samples from a uniform mixture of $D_1, \ldots, D_k$. We will prove that our algorithm succeeds under the following condition on the samples $X_1, \ldots, X_n$.

(D1) (Empirical moments) For every cluster $S_j = \{X_i : X_i \text{ is from } D_j\}$, the system $\widehat{\mathcal{A}}$ from Lemma 4.4.1 with $\alpha = 1/m$ and $\tau = \Delta^{-t}$ has a solution which takes $w \in \{0,1\}^n$ to be the 0/1 indicator vector of S_j.

(D2) (Empirical means) Let $\overline{\mu}_j$ be the empirical mean of cluster S_j. The $\overline{\mu}_j$'s satisfy $\|\overline{\mu}_i - \mu_i\|_2 \leq \Delta^{-t}$.

We note a few useful consequences of these conditions, especially (D1). First of all, it implies all clusters have almost the same size: $(1 - \Delta^{-t}) \cdot \frac{n}{k} \leq |S_j| \leq (1 + \Delta^{-t}) \cdot \frac{n}{k}$. Second, it implies that all clusters have explicitly bounded moments: for every S_j,

$$\vdash_t \ \frac{k}{n} \sum_{i \in S_j} \langle X_i - \overline{\mu}_j, u \rangle^t \leq 2 \cdot t^{t/2} \cdot \|u\|_2^t \ .$$

Lemmas The following key lemma captures our SoS identifiability proof for mixture models.

Lemma 4.5.3. *Let $\mu_1, \ldots, \mu_k, D_1, \ldots, D_k$ be as in Theorem 4.5.1, with mean separation Δ. Suppose (D1), (D2) occur for samples $X_1, \ldots, X_n$. Let $t \in \mathbb{N}$ be a power of two. Let $\tilde{\mathbb{E}}$ be a degree-$O(t)$ pseudoexpectation which satisfies $\mathcal{A}$ from Lemma 4.4.1 with $\alpha = 1/k$ and $\tau \leq \Delta^{-t}$. Then for every $j, \ell \in [k]$,*

$$\tilde{\mathbb{E}} \langle a_j, w \rangle \langle a_\ell, w \rangle \leq 2^{8t+8} \cdot t^{t/2} \cdot \frac{n^2}{k} \cdot \frac{1}{\Delta^t} \ .$$

LearnMixtureMeans we can assume that $\|\mu_i\|_2 \leq \mathrm{poly}(d,k)$ for every $i \leq d$.

The other main lemma shows that conditions (D1) and (D2) occur with high probability.

Lemma 4.5.4 (Concentration for mixture models)**.** *With notation as above, conditions (D1) and (D2) simultaneously occur with probability at least* $1 - 1/d^{15}$ *over samples* $X_1, \ldots, X_n$*, so long as* $n \geq d^{O(t)} k^{O(1)}$*, for* $\Delta \geq 1$*.*

Lemma 4.5.4 follows from Lemma 4.4.1, for (D1), and standard concentration arguments for (D2). Now we can prove the main theorem.

Proof of Theorem 4.5.1 (uniform mixtures case). Suppose conditions (D1) and (D2) hold. Our goal will be to bound $\|M - A\|_F^2 \leq n \cdot \frac{2^{O(t)} t^{t/2} k^4}{\Delta^t}$, where A is the 0/1 indicator matrix for the ground truth partition $S_1, \ldots, S_k$ of $X_1, \ldots, X_n$ according to $D_1, \ldots, D_k$. Then by Lemma 4.5.2, the rounding algorithm will return a partition $C_1, \ldots, C_k$ of $[n]$ such that C_ℓ and S_ℓ differ by at most $n \frac{2^{O(t)} t^{t/2} k^{10}}{\Delta^t}$ points, with probability at least $1 - \frac{2^{O(t)} t^{t/2} k^{30}}{\Delta^t}$. By the guarantees of Theorem 4.6.1 regarding the algorithm EstimateMean, with high probability the resulting error in the mean estimates $\hat{\mu}_i$ will satisfy

$$\|\mu_i - \hat{\mu}_i\|_2 \leq \sqrt{t} \cdot \left(\frac{2^{O(t)} t^{t/2} k^{10}}{\Delta^t} \right)^{\frac{t-1}{t}} \leq \frac{2^{O(t)} \cdot t^{t/2} \cdot k^{10}}{\Delta^{t-1}} \,.$$

We turn to the bound on $\|M - A\|_F^2$. First we bound $\langle \tilde{\mathbb{E}}\, ww^\top, A \rangle$. Getting started,

$$\tilde{\mathbb{E}} \left(\sum_{i \in [k]} \langle w, a_i \rangle \right)^2 = \tilde{\mathbb{E}} \left(\sum_{i \in [n]} w_i \right)^2 \geq (1 - \Delta^{-t})^2 \cdot n^2 / k^2 \,.$$

By Lemma 4.5.3, choosing t later,

$$\sum_{i \neq j \in [k]} \tilde{\mathbb{E}} \langle a_i, w \rangle \langle a_j, w \rangle \leq n^2 2^{O(t)} t^{t/2} \cdot k \cdot \frac{1}{\Delta^t} \,.$$

Together, these imply

$$\tilde{\mathbb{E}} \sum_{i \in [k]} \langle w, a_i \rangle^2 \geq \frac{n^2}{k^2} \cdot \left[1 - \frac{2^{O(t)} t^{t/2} k^3}{\Delta^t} \right] \,.$$

At the same time, $\| \tilde{\mathbb{E}} ww^T \|_F \leq \frac{1}{k} \|A\|_F$ by minimality (since the uniform distribution over cluster indicators satisfies $\mathcal{A}$), and by routine calculation and assumption (D1), $\|A\|_F \leq \frac{n}{\sqrt{k}}(1 + O(\Delta^{-t}))$. Together, we have obtained

$$\langle M, A \rangle \geq \left(1 - \frac{2^{O(t)} t^{t/2} k^3}{\Delta^t}\right) \cdot \|A\|_F \|M\|_F$$

which can be rearranged to give $\|M - A\|_F^2 \leq n \cdot \frac{2^{O(t)} t^{t/2} k^4}{\Delta^t}$. □

4.5.3 Identifiability

In this section we prove Lemma 4.5.3. We use the following helpful lemmas. The first is in spirit an SoS version of Lemma 4.2.1.

Lemma 4.5.5. *Let $\mu_1, \ldots, \mu_k, D_1, \ldots, D_k, t$ be as in Theorem 4.5.1. Let $\overline{\mu}_i$ be as in (D1). Suppose (D1) occurs for samples $X_1, \ldots, X_n$. Let $\mathcal{A}$ be the system from Lemma 4.4.1, with $\alpha = 1/k$ and any τ. Then*

$$\mathcal{A} \vdash_{O(t)} \langle a_j, w \rangle^t \|\mu - \overline{\mu}_j\|_2^{2t} \leq 2^{t+2} t^{t/2} \cdot \frac{n}{k} \cdot \langle a_j, w \rangle^{t-1} \cdot \|\mu - \overline{\mu}_j\|_2^t \, .$$

The second lemma is an SoS triangle inequality, capturing the consequences of separation of the means. The proof is standard given Fact D.1.2.

Lemma 4.5.6. *Let $a, b \in \mathbb{R}^k$ and $t \in \mathbb{N}$ be a power of 2. Let $\Delta = \|a - b\|_2$. Let $u = (u_1, \ldots, u_k)$ be indeterminates. Then $\vdash_t \|a - u\|_2^t + \|b - u\|_2^t \geq 2^{-t} \cdot \Delta^t$.*

The last lemma helps put the previous two together. Although we have phrased this lemma to concorde with the mixture model setting, we note that the proof uses nothing about mixture models and consists only of generic manipulations of pseudodistributions.

Lemma 4.5.7. *Let $\mu_1, \ldots, \mu_k, D_1, \ldots, D_k, X_1, \ldots, X_n$ be as in Theorem 4.5.1. Let a_j be the 0/1 indicator for the set of samples drawn from D_j. Suppose $\tilde{\mathbb{E}}$ is a degree-$O(t)$*

pseudodistribution which satisfies

$$\begin{aligned}\langle a_j, w\rangle &\le n\\ \langle a_\ell, w\rangle &\le n\\ \|\mu-\overline{\mu}_j\|_2^{2t}+\|\mu-\overline{\mu}_\ell\|_2^{2t} &\ge A\\ \langle a_j, w\rangle^t\|\mu-\overline{\mu}_j\|_2^{2t} &\le Bn\langle a_j, w\rangle^{t-1}\|\mu-\overline{\mu}_j\|_2^t\\ \langle a_\ell, w\rangle^t\|\mu-\overline{\mu}_\ell\|_2^{2t} &\le Bn\langle a_\ell, w\rangle^{t-1}\|\mu-\overline{\mu}_\ell\|_2^t\end{aligned}$$

for some scalars $A, B \ge 0$*. Then*

$$\tilde{\mathbb{E}}\langle a_j, w\rangle\langle a_\ell, w\rangle \le \frac{2n^2B}{\sqrt{A}}\ .$$

Now we have the tools to prove Lemma 4.5.3.

Proof of Lemma 4.5.3. We will verify the conditions to apply Lemma 4.5.7. By Lemma 4.5.5, when (D1) holds, the pseudoexpectation $\tilde{\mathbb{E}}$ satisfies

$$\langle a_j, w\rangle^t\|\mu-\overline{\mu}_j\|_2^{2t} \quad \le Bn\langle a_j, w\rangle^{t-1}\|\mu-\overline{\mu}_j\|_2^t$$

for $B = 4(4t)^{t/2}/k$, and similarly with j, ℓ interposed. Similarly, by separation of the empirical means, $\tilde{\mathbb{E}}$ satisfies $\|\mu-\overline{\mu}_j\|_2^{2t}+\|\mu-\overline{\mu}_\ell\|_2^{2t} \ge A$ for $A = 2^{-2t}\Delta^{2t}$, recalling that the empirical means are pairwise separated by at least $\Delta - 2\Delta^{-t}$. Finally, clearly $\mathcal{A} \vdash_{O(1)} \langle a_j, w\rangle \le n$ and similarly for $\langle a_\ell, w\rangle$. So applying Lemma 4.5.7 we get

$$\tilde{\mathbb{E}}\langle a_j, w\rangle\langle a_\ell, w\rangle \le \frac{2n^2B}{\sqrt{A}} \le \frac{n^2 2^{2t+2}t^{t/2}}{k}\cdot\frac{1}{\Delta^t}\ . \quad \square$$

4.5.4 Proof of Lemma 4.5.5

In this subsection we prove Lemma 4.5.5. We use the following helpful lemmata. The first bounds error from samples selected from the wrong cluster using the moment inequality.

Lemma 4.5.8. *Let $j, \mathcal{A}, X_1, \ldots, X_n, \mu_j, \overline{\mu}_j$ be as in Lemma 4.5.5. Then*

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \langle \mu - X_i, \mu - \overline{\mu}_j \rangle \right)^t \leq 2t^{t/2} \cdot \langle a_j, w \rangle^{t-1} \| \mu - \overline{\mu}_j \|_2^t \,.$$

Proof. The proof goes by Hölder's inequality followed by the moment inequality in $\mathcal{A}$. Carrying this out, by Fact D.1.6 and evenness of t,

$$\{w_i^2 = w_i\} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \langle \mu - X_i, \mu - \overline{\mu}_j \rangle \right)^t \leq \left(\sum_{i \in S_j} w_i \right)^{t-1} \cdot \left(\sum_{i \in [n]} w_i \langle \mu - X_i, \mu - \overline{\mu}_j \rangle^t \right) \,.$$

Then, using the main inequality in $\mathcal{A}$,

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \right)^{t-1} \cdot 2t^{t/2} \cdot \| \mu - \overline{\mu}_j \|_2^t = 2t^{t/2} \cdot \langle a_j, w \rangle^{t-1} \| \mu - \overline{\mu}_j \|_2^t \,. \quad \square$$

The second lemma bounds error from deviations in the empirical t-th moments of the samples from the j-th cluster.

Lemma 4.5.9. *Let $\mu_1, \ldots, \mu_k, D_1, \ldots, D_k$ be as in Theorem 4.5.1. Suppose condition (D1) holds for samples $X_1, \ldots, X_n$. Let $w_1, \ldots, w_n$ be indeterminates. Let $u = u_1, \ldots, u_d$ be an indeterminate. Then for every $j \in [k]$,*

$$\{w_i^2 = w_i\} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \langle X_i - \overline{\mu}_j, u \rangle \right)^t \leq \langle a_j, w \rangle^{t-1} \cdot 2 \cdot \frac{n}{k} \cdot \|u\|_2^t \,.$$

Proof. The first step is Hölder's inequality again:

$$\{w_i^2 = w_i\} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \langle X_i - \overline{\mu}_j, u \rangle \right)^t \leq \langle a_j, w \rangle^{t-1} \cdot \sum_{i \in S_j} \langle X_i - \overline{\mu}_j, u \rangle^t \,.$$

Finally, condition (D1) yields

$$\{w_i^2 = w_i\} \vdash_{O(t)} \left(\sum_{i \in S_j} w_i \langle X_i - \overline{\mu}_j, u \rangle \right)^t \leq \langle a_j, w \rangle^{t-1} \cdot 2 \cdot \frac{n}{k} \cdot \|u\|_2^t \,. \quad \square$$

We can prove Lemma 4.5.5 by putting together Lemma 4.5.8 and Lemma 4.5.9.

Proof of Lemma 4.5.5. Let $j \in [k]$ be a cluster and recall $a_j \in \{0,1\}^n$ is the 0/1 indicator for the samples in cluster j. Let S_j be the samples in the j-th cluster, with empirical mean $\overline{\mu}_j$. We begin by writing $\langle a_j, w \rangle \|\mu - \overline{\mu}_j\|_2^2$ in terms of samples $X_1, \ldots, X_n$.

$$\begin{aligned} \langle a_j, w \rangle \|\mu - \overline{\mu}_j\|_2^2 &= \sum_{i \in [n]} w_i \langle \mu - \overline{\mu}_j, \mu - \overline{\mu}_j \rangle \\ &= \sum_{i \in S_j} w_i \langle \mu - X_i, \mu - \overline{\mu}_j \rangle + \sum_{i \in [n]} w_i \langle X_i - \overline{\mu}_j, \mu - \overline{\mu}_j \rangle \,. \end{aligned}$$

Hence, using $(a+b)^t \leq 2^t(a^t + b^t)$, we obtain

$$\vdash_{O(t)} \langle a_j, w \rangle^t \|\mu - \overline{\mu}_j\|_2^{2t} \leq 2^t \cdot \left(\sum_{i \in S_j} w_i \langle \mu - X_i, \mu - \overline{\mu}_j \rangle \right)^t + 2^t \cdot \left(\sum_{i \in S_j} w_i \langle X_i - \overline{\mu}_j, \mu - \overline{\mu}_j \rangle \right)^t \,.$$

Now using Lemma 4.5.8 and Lemma 4.5.9,

$$\mathcal{A} \vdash_{O(t)} \langle a_j, w \rangle^t \|\mu - \overline{\mu}_j\|_2^{2t} \leq 2^{t+2} t^{t/2} \cdot \frac{n}{k} \cdot \langle a_j, w \rangle^{t-1} \cdot \|\mu - \overline{\mu}_j\|_2^t$$

as desired. □

4.5.5 Proof of Lemma 4.5.7

We prove Lemma 4.5.7. The proof only uses standard SoS and pseudodistribution tools. The main inequality we will use is the following version of Hölder's inequality.

Fact 4.5.10 (Pseudoexpectation Hölder's, see Lemma A.4 in [BKS14]). *Let p be a degree-ℓ polynomial. Let $t \in N$ and let $\tilde{\mathbb{E}}$ be a degree-$O(t\ell)$ pseudoexpectation on* *indeterminates x. Then*

$$\tilde{\mathbb{E}}\, p(x)^{t-2} \le \left(\tilde{\mathbb{E}}\, p(x)^t\right)^{\frac{t-2}{t}} \, .$$

Now we can prove Lemma 4.5.7.

Proof of Lemma 4.5.7. We first establish the following inequality.

$$\tilde{\mathbb{E}}\langle a_j, w\rangle^t \langle a_\ell, w\rangle^t \|\mu - \overline{\mu}_j\|_2^{2t} \le B^2 n^2 \cdot \tilde{\mathbb{E}}\langle a_j, w\rangle^{t-2}\langle a_\ell, w\rangle^t \, . \tag{4.4}$$

(The inequality will also hold by symmetry with j and ℓ exchanged.) This we do as follows:

$$\begin{aligned}\tilde{\mathbb{E}}\langle a_j, w\rangle^t \langle a_\ell, w\rangle^t \|\mu - \overline{\mu}_j\|_2^{2t} &\le Bn\, \tilde{\mathbb{E}}\langle a_j, w\rangle^{t-1}\langle a_\ell, w\rangle^t \|\mu - \overline{\mu}_j\|_2^t \\ &\le Bn\left(\tilde{\mathbb{E}}\langle a_j, w\rangle^{t-2}\langle a_\ell, w\rangle^t\right)^{1/2} \cdot \left(\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t \|\mu - \overline{\mu}_j\|_2^{2t}\right)^{1/2}\end{aligned}$$

where the first line is by assumption on $\tilde{\mathbb{E}}$ and the second is by pseudoexpectation Cauchy-Schwarz. Rearranging gives the inequality (4.4).

Now we use this to bound $\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t$. By hypothesis,

$$\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t \le \frac{1}{A}\, \tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t(\|\mu - \overline{\mu}_j\|_2^{2t} + \|\mu - \overline{\mu}_\ell\|_2^{2t}) \, ,$$

which, followed by (4.4) gives

$$\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t \le \frac{1}{A} \cdot B^2 n^2 \cdot \tilde{\mathbb{E}}\left[\langle a_j, w\rangle^{t-2}\langle a_\ell, w\rangle^t + \langle a_\ell, w\rangle^{t-2}\langle a_j, w\rangle^t\right] \, .$$

Using $\langle a_j, w\rangle, \langle a_\ell, w\rangle \le n$, we obtain

$$\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t \le \frac{2}{A} \cdot B^2 n^4 \cdot \tilde{\mathbb{E}}\langle a_j, w\rangle^{t-2}\langle a_\ell, w\rangle^{t-2} \, .$$

Finally, using Fact 4.5.10, the right side is at most $2B^2n^4/A \cdot \left(\tilde{\mathbb{E}}\langle a_j, w\rangle^t\langle a_\ell, w\rangle^t\right)^{(t-2)/t}$, so cancelling terms we get

$$\left(\tilde{\mathbb{E}}\langle a_j, w\rangle^t \langle a_\ell, w\rangle^t\right)^{2/t} \leq \frac{2B^2 n^4}{A} \,.$$

Raising both sides to the $t/2$ power gives

$$\tilde{\mathbb{E}}\langle a_j, w\rangle^t \langle a_\ell, w\rangle^t \leq \frac{2^{t/2} B^t n^{2t}}{A^{t/2}} \,,$$

and finally using Cauchy-Schwarz,

$$\tilde{\mathbb{E}}\langle a_j, w\rangle \langle a_\ell, w\rangle \leq \left(\tilde{\mathbb{E}}\langle a_j, w\rangle^t \langle a_\ell, w\rangle^t\right)^{1/t} \leq \frac{2n^2 B}{\sqrt{A}} \,. \quad \square$$

4.5.6 Rounding

In this section we state and analyze our second-moment round algorithm. As have discussed already, our SoS proofs in the mixture model setting are quite strong, meaning that the rounding algorithm is relatively naive.

The setting in this section is as follows. Let $n, m \in \mathbb{N}$ with $m \ll n$. There is a ground-truth partition of $[n]$ into m parts $S_1, \ldots, S_m$ such that $|S_i| = (1 \pm \delta)\frac{n}{m}$. Let $A \in \mathbb{R}^{n \times n}$ be the 0/1 indicator matrix for this partition, so $A_{ij} = 1$ if $i, j \in S_\ell$ for some ℓ and is 0 otherwise. Let $M \in \mathbb{R}^{n \times n}$ be a matrix such that $\|M - A\|_F \leq \varepsilon n$. The algorithm takes M and outputs a partition $C_1, \ldots, C_m$ of $[m]$ which makes few errors compared to $S_1, \ldots, S_m$.

We will prove the following theorem.

Theorem 4.5.11. *With notation as before Algorithm 12 with $E = m$, with probability at least $1 - \varepsilon^2 m^3$ Algorithm 12 returns a partition $C_1, \ldots, C_m$ of $[n]$ such that (up to a permutation of $[m]$), $C_\ell = T_\ell \cup B_\ell$, where $T_\ell \subseteq S_\ell$ has size $|T_\ell| \geq |S_\ell| - \varepsilon^2 mn$ and $|B_\ell| \leq \varepsilon^2 mn$.*

To get started analyzing the algorithm, we need a definition.

Definition 4.5.1. For cluster S_j, let $a_j \in \mathbb{R}^n$ be its 0/1 indicator vector. If $i \in S_j$, we say it is *E-good* if $\|v_i - a_j\|_2 \leq \sqrt{n/E}$, and otherwise *$E$-bad*, where v_i is the i-th row

Algorithm 12 Rounding the second moment of $\tilde{\mathbb{E}}[ww^\top]$

```
function RoundSecondMoments($M \in \mathbb{R}^{n \times n}, E \in \mathbb{R}$)
    Let $S = [n]$
    Let $v_1, \ldots, v_n$ be the rows of $M$
    for $\ell = 1, \ldots, m$ do
        Choose $i \in S$ uniformly at random
        Let
            $C_\ell = \left\{ i' \in S : \|v_i - v_{i'}\|_2 \leq 2\frac{n^{1/2}}{E} \right\}$
        Let $S \leftarrow S \setminus C_\ell$
    return The clusters $C_1, \ldots, C_m$.
```

of M. Let $I_g \subseteq [n]$ denote the set of E-good indices and I_b denote the set of E-bad indices. (We will choose E later.) For any $j = 1, \ldots, k$, let $I_{g,j} = I_g \cap S_j$ denote the set of good indices from cluster j.

We have:

Lemma 4.5.12. *Suppose E as in* RoundSecondMoments *satisfies $E \geq m/8$. Suppose that in iterations $1, \ldots, m$,* RoundSecondMoments *has chosen only good vectors. Then, there exists a permutation $\pi : [m] \to [m]$ so that $C_\ell = I_{g,\pi(\ell)} \cup B_\ell$, where $B_\ell \subseteq I_b$ for all ℓ.*

Proof. We proceed inductively. We first prove the base case. WLOG assume that the algorithm picks v_1, and that v_1 is good, and is from component j. Then, for all $i \in I_{g,j}$, by the triangle inequality we have $\|v_i - v_1\|_2 \leq 2\frac{n^{1/2}}{B}$, and so $I_{g,j} \subseteq C_1$. Moreover, if $i \in I_{g,j'}$ for some $j' \neq j$, we have

$$\|v_i - v_1\|_2 \geq \|a_j' - a_j\|_2 - 2\frac{n^{1/2}}{E^{1/2}} \geq \frac{n^{1/2}}{\sqrt{m}} - 2\frac{n^{1/2}}{E^{1/2}} > 2\frac{n^{1/2}}{E^{1/2}},$$

and so in this case $i \notin C_1$. Hence $C_1 = I_{g,j} \cup B_1$ for some $B_1 \subseteq I_b$.

Inductively, suppose that if the algorithm chooses good indices in iterations $1, \ldots, a-1$, then there exist distinct $j_1, \ldots, j_{a-1}$ so that $C_\ell = I_{g,j_\ell} \cup B_\ell$ for $B_\ell \subseteq I_b$. We seek to prove that if the algorithm chooses a good index in iteration a, then $C_a = I_{g,j_a} \cup B_a$ for some $j_a \notin \{j_1, \ldots, j_{a-1}\}$ and $B_a \subseteq I_b$. Clearly by induction this proves the Lemma.

WLOG assume that the algorithm chooses v_1 in iteration a. Since by assumption 1 is good, and we have removed I_{g_ℓ} for $\ell = 1, \ldots, a-1$, then $1 \in I_{g,j_a}$ for some $j_a \notin \{j_1, \ldots, j_{a-1}\}$. Then, the conclusion follows from the same calculation as in the base case. □

Lemma 4.5.13. *There are at most $\varepsilon^2 En$ indices which are E-bad; i.e. $|I_b| \leq \varepsilon^2 En$.*

Proof. We have

$$\begin{aligned}\varepsilon^2 n^2 &\geq \left\| M - \sum_{i \leq m} a_i a_i^\top \right\|_F^2 \geq \sum_j \sum_{i \in S_j \text{ bad}} \|v_i - a_j\|_2^2 \\ &\geq \frac{n}{E} |I_b| \ ,\end{aligned}$$

from which the claim follows by simplifying. □

This in turns implies:

Lemma 4.5.14. *With probability at least $1 - \varepsilon^2 m^3$, the algorithm* RoundSecondMoments *chooses good indices in all k iterations.*

Proof. By Lemma 4.5.13, in the first iteration the probability that a bad vector is chosen is at most $\varepsilon^2 E$. Conditioned on the event that in iterations $1, \ldots, a$ the algorithm has chosen good vectors, then by Lemma 4.5.12, there is at least one j_a so that no points in I_{g,j_a} have been removed. Thus at least $(1-\delta)n/m$ vectors remain, and in total there are at most $\varepsilon^2 En$ bad vectors, by Lemma 4.5.13. So, the probability of choosing a bad vector is at most $\varepsilon^2 Em$. Therefore, by the chain rule of conditional expectation and our assumption , the probability we never choose a bad vector is at least

$$\left(1 - \varepsilon^2 Em\right)^m$$

Choosing $E = m$ this is $(1 - \varepsilon^2 m^2)^m \geq 1 - \varepsilon^2 m^3$. as claimed. □

Now Theorem 4.5.11 follows from putting together the lemmas.

4.6 Robust estimation: algorithm and analysis

Our algorithm for robust estimation is very similar to our algorithm for mixture models. Suppose the underlying distribution D, whose mean μ^* the algorithm robustly estimates, is $10t$-explicitly bounded. As a reminder, the input to the algorithm is a list of $X_1, \ldots, X_n \in \mathbb{R}^d$ and a sufficiently-small $\varepsilon > 0$. The guarantee is that at least $(1-\varepsilon)n$ of the vectors were sampled according to D, but εn of the vectors were chosen adversarially.

The algorithm solves a semidefinite program to obtain a degree $O(t)$ pseudodistribution which satisfies the system $\mathcal{A}$ from Section 4.4 with $\alpha = 1 - \varepsilon$ and $\tau = 0$. Throughout this section, we will always assume that $\mathcal{A}$ is instantiated with these parameters, and omit them for conciseness. Then the algorithm just outputs $\tilde{\mathbb{E}}\,\mu$ as its estimator for μ^*.

Our main contribution in this section is a formal description of an algorithm EstimateMean which makes these ideas rigorous, and the proof of the following theorem about its correctness:

Theorem 4.6.1. *Let $\varepsilon > 0$ sufficiently small and $t \in \mathbb{N}$. Let D be a $10t$-explicitly bounded distribution over $\mathbb{R}^d$ with mean μ^*. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from D where $n = d^{O(t)}/\varepsilon^2$. Then, given ε, t and $X_1, \ldots, X_n$, the algorithm* EstimateMean *runs in time $d^{O(t)}$ and outputs μ so that $\|\mu - \mu^*\|_2 \le O(t^{1/2}\varepsilon^{1-1/t})$, with probability at least $1 - 1/d$.*

As a remark, observe that if we set $t = 2\log 1/\varepsilon$, then the error becomes $O(\varepsilon\sqrt{\log 1/\varepsilon})$. Thus, with $n = O(d^{O(\log 1/\varepsilon)}/\varepsilon^2)$ samples and $n^{O(\log 1/\varepsilon)} = d^{O(\log 1/\varepsilon)^2}$ runtime, we achieve the same error bounds for general explicitly bounded distributions as the best known polynomial time algorithms achieve for Gaussian mean estimation.

4.6.1 Additional Preliminaries

Throughout this section, let $[n] = S_{\text{good}} \cup S_{\text{bad}}$, where S_{good} is the indices of the uncorrupted points, and S_{bad} is the indices of the corrupted points, so that $|S_{\text{bad}}| = \varepsilon n$ by assumption. Moreover, let $Y_1, \ldots, Y_n$ be iid from D so that $Y_i = X_i$ for all $i \in S_{\text{good}}$.

We now state some additional tools we will require in our algorithm.

Naive Pruning We will require the following elementary pruning algorithm, which removes all points which are very far away from the mean. We require this only to avoid some bit-complexity issues in semidefinite programming; in particular we just need to ensure that the vectors $X_1, \ldots, X_n$ used to form the SDP have polynomially-bounded norms. Formally:

Lemma 4.6.2 (Naive pruning). *Let ε, t, μ^*, and $X_1, \ldots, X_n$ be as in Theorem 4.6.1. There is an algorithm* NaivePrune*, which given ε, t and $X_1, \ldots, X_n$, runs in time $O(\varepsilon d n^2)$, and outputs a subset $S \subseteq [n]$ so that with probability $1 - 1/d^{10}$, the following holds:*

- *No uncorrupted points are removed, that is $S_{\text{good}} \subseteq S$, and*
- *For all $i \in S$, we have $\|X_i - \mu^*\|_2 \leq O(d)$.*

In this case, we say that NaivePrune *succeeds.*

This algorithm goes by straightforward outlier-removal. It is very similar to the procedure described in Fact 2.2.6 (using bounded t-th moments instead of sub-Gaussianity), so we omit it.

Satisfiability In our algorithm, we will use the same set of polynomial equations $\widehat{\mathcal{A}}$ as in Lemma 4.4.1. However, the data we feed in does not exactly fit the assumptions in the Lemma. Specifically, because the adversary is allowed to remove an ε-fraction of good points, the resulting uncorrupted points are no longer iid from D. Despite this, we are able to specialize Lemma 4.4.1 to this setting:

Lemma 4.6.3. *Fix $\varepsilon > 0$ sufficiently small, and let $t \in \mathbb{N}, t \geq 4$ be a power of 2. Let D be a $10t$-explicitly bounded distribution. Let $X_1, \ldots, X_n \in \mathbb{R}^d$ be an ε-corrupted set of samples from D, and let $\widehat{\mathcal{A}}$ be as in Lemma 4.4.1. The conclusion (1 – Satisfiability) of Lemma 4.4.1 holds, with w taken to be the $0/1$ indicator of the $(1 - \varepsilon)n$ good samples among $X_1, \ldots, X_n$.*

We sketch the proof of Lemma 4.6.3 in Section 4.7.4.

4.6.2 Formal Algorithm Specification

With these tools in place, we can now formally state the algorithm. The formal specification of this algorithm is given in Algorithm 13.

Algorithm 13 Robust Mean Estimation

```
function EstimateMean($\varepsilon, t, \kappa, X_1, \dots, X_n$)
    Preprocess: let $X_1, \dots, X_n \leftarrow$ NaivePrune($\varepsilon, X_1, \dots, X_n$), and let $\widehat{\mu}$ be the empirical mean
    Let $X_i \leftarrow X_i - \widehat{\mu}$
    By semidefinite programming, find a pseudoexpectation of degree $O(t)$ which satisfies the structured subset polynomials from Lemma 4.6.3, with $\alpha = (1-\varepsilon)n$ and $\tau = 0$.
    return $\tilde{\mathbb{E}}\,\mu + \widehat{\mu}$.
```

The first two lines of Algorithm 13 are only necessary for bit complexity reasons, since we cannot solve SDPs exactly. However, since we can solve them to doubly-exponential accuracy in polynomial time, it suffices that all the quantities are at most polynomially bounded (indeed, exponentially bounded suffices) in norm, which these two lines easily achieve. For the rest of this section, for simplicity of exposition, we will ignore these issues.

4.6.3 Deterministic conditions

With these tools in place, we may now state the deterministic conditions under which our algorithm will succeed. Throughout this section, we will condition on the following events holding simultaneously:

(E1) NaivePrune succeeds,

(E2) The conclusion of Lemma 4.6.3 holds,

(E3) We have the following concentration of the uncorrupted points:

$$\left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i - \mu^* \right\|_2 \le O(t^{1/2} \varepsilon^{1-1/t}) \ , \text{ and}$$

(E4) We have the following concentration of the empirical t-th moment tensor:

$$\frac{1}{n}\sum_{i\in[n]}\left[(Y_i-\mu^*)^{\otimes t/2}\right]\left[(Y_i-\mu^*)^{\otimes t/2}\right]^\top \preceq \mathop{\mathbb{E}}_{X\sim D}\left[(X-\mu^*)^{\otimes t/2}\right]\left[(X-\mu^*)^{\otimes t/2}\right]^\top + 0.1\cdot I\ ,$$

for I is the $d^{t/2}\times d^{t/2}$-sized identity matrix.

The following lemma says that with high probability, these conditions hold simultaneously:

Lemma 4.6.4. *Let ε, t, μ^*, and $X_1,\ldots,X_n\in\mathbb{R}^d$ be as in Theorem 4.6.1. Then, Conditions (E1)-(E4) hold simultaneously with probability at least $1-1/d^5$.*

We defer the proof of this lemma to the Appendix.

For simplicity of notation, throughout the rest of the section, we will assume that NaivePrune does not remove any points whatsoever. Because we are conditioning on the event that it removes no uncorrupted points, it is not hard to see that this is without loss of generality.

4.6.4 Identifiability

Our main identifiability lemma is the following.

Lemma 4.6.5. *Let ε, t, μ^* and $X_1,\ldots,X_n\in\mathbb{R}^d$ be as in Theorem 4.6.1, and suppose they satisfy (E1)–(E4). Then, we have*

$$\mathcal{A}\vdash_{O(t)} \|\mu-\mu^*\|_2^{2t}\le O(t^{t/2})\cdot\varepsilon^{t-1}\cdot\|\mu-\mu^*\|_2^t\ .$$

Since this lemma is the core of our analysis for robust estimation, in the remainder of this section we prove it. The proof uses the following three lemmas to control three sources of error in $\tilde{\mathbb{E}}\,\mu$, which we prove in Section 4.6.6. The first, Lemma 4.6.6 controls sampling error from true samples from D.

Lemma 4.6.6. *Let ε, t, μ^* and $X_1, \ldots, X_n \in \mathbb{R}^d$ be as in Theorem 4.6.1, and suppose they satisfy (E1)–(E4). Then, we have*

$$\vdash_{O(t)} \left(\sum_{i \in S_{\text{good}}} \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t \leq O(\varepsilon^{t-1}) \cdot t^{t/2} \cdot n^t \cdot \|\mu - \mu^*\|_2^t \,.$$

To describe the second and third error types, we think momentarily of $w \in \mathbb{R}^n$ as the $0/1$ indicator for a set S of samples whose empirical mean will be the output of the algorithm. (Of course this is not strictly true, but this is a convenient mindset in constructing SoS proofs.) The second type of error comes from the possible failure of S to capture some ε fraction of the good samples from D. Since D has $O(t)$ bounded moments, if T is a set of m samples from D, the empirical mean of any $(1-\varepsilon)m$ of them is at most $\varepsilon^{1-1/t}$-far from the true mean of D.

Lemma 4.6.7. *Let ε, t, μ^* and $X_1, \ldots, X_n \in \mathbb{R}^d$ be as in Theorem 4.6.1, and suppose they satisfy (E1)–(E4). Then, we have*

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_{\text{good}}} (w_i - 1) \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t \leq 2\varepsilon^{t-1} n^t \cdot t^{t/2} \cdot \|\mu - \mu^*\|_2^t \,.$$

The third type of error is similar in spirit: it is the contribution of the original uncorrupted points that the adversary removed. Formally:

Lemma 4.6.8. *Let ε, t, μ^* and $X_1, \ldots, X_n \in \mathbb{R}^d$ and $Y_1, \ldots, Y_n \in \mathbb{R}^d$ be as in Theorem 4.6.1, and suppose they satisfy (E1)–(E4). Then, we have*

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_{\text{bad}}} \langle Y_i - \mu^*, \mu - \mu^* \rangle \right)^t \leq 2\varepsilon^{t-1} n^t \cdot t^{t/2} \cdot \|\mu - \mu^*\|_2^t \,.$$

Finally, the fourth type of error comes from the εn adversarially-chosen vectors. We prove this lemma by using the bounded-moments inequality in $\mathcal{A}$.

Lemma 4.6.9. *Let ε, t, μ^* and $X_1, \ldots, X_n \in \mathbb{R}^d$ be as in Theorem 4.6.1, and suppose* *they satisfy (E1)–(E4). Then, we have*

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \notin S_{\text{good}}} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t \leq 2\varepsilon^{t-1} n^t \cdot t^{t/2} \cdot \|\mu - \mu^*\|_2^t \,.$$

With these lemmas in place, we now have the tools to prove Lemma 4.6.5.

Proof of Lemma 4.6.5. Let $Y_1, \ldots, Y_n \in \mathbb{R}^d$ be as in Theorem 4.6.1. We expand the norm $\|\mu - \mu^*\|_2^2$ as $\langle \mu - \mu^*, \mu - \mu^* \rangle$ and rewrite $\sum_{i \in [n]} w_i \mu$ as $\sum_{i \in [n]} w_i X_i$:

$$\begin{aligned}
\sum_{i \in [n]} w_i \|\mu - \mu^*\|_2^2 &\overset{(a)}{=} \sum_{i \in [n]} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle \\
&\overset{(b)}{=} \sum_{i \in S_{\text{good}}} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle + \sum_{i \in S_{\text{bad}}} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle \\
&\overset{(c)}{=} \sum_{i \in S_{\text{good}}} \langle X_i - \mu^*, \mu - \mu^* \rangle + \sum_{i \in S_{\text{good}}} (w_i - 1) \langle X_i - \mu^*, \mu - \mu^* \rangle \\
&\quad + \sum_{i \in S_{\text{bad}}} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle \\
&\overset{(d)}{=} \sum_{i \in [n]} \langle X_i - \mu^*, \mu - \mu^* \rangle + \sum_{i \in S_{\text{good}}} (w_i - 1) \langle X_i - \mu^*, \mu - \mu^* \rangle \\
&\quad - \sum_{i \in S_{\text{bad}}} \langle Y_i - \mu^*, \mu - \mu^* \rangle + \sum_{i \in S_{\text{bad}}} w_i \langle X_i - \mu^*, \mu - \mu^* \rangle \,,
\end{aligned}$$

where (a) follows from the mean axioms, (b) follows from splitting up the uncorrupted and the corrupted samples, (c) follows by adding and subtracting 1 to each term in S_{good}, and (d) follows from the assumption that $Y_i = X_i$ for all $i \in [n]$. We will rearrange the last term by adding and subtracting μ. Note the following polynomial identity:

$$\langle X_i - \mu^*, \mu - \mu^* \rangle = \langle X_i - \mu, \mu - \mu^* \rangle + \|\mu - \mu^*\|_2^2$$

and put it together with the above to get

$$\sum_{i\in[n]} w_i\|\mu-\mu^*\|_2^2 = \sum_{i\in S_{\text{good}}} \langle X_i-\mu^*,\mu-\mu^*\rangle + \sum_{i\in S_{\text{good}}} (w_i-1)\langle X_i-\mu^*,\mu-\mu^*\rangle$$
$$- \sum_{i\in S_{\text{bad}}} \langle Y_i-\mu^*,\mu-\mu^*\rangle + \sum_{i\in S_{\text{bad}}} w_i\langle X_i-\mu,\mu-\mu^*\rangle + \sum_{i\in S_{\text{bad}}} w_i\|\mu-\mu^*\|_2^2 \,.$$

which rearranges to

$$\sum_{i\in S_{\text{good}}} w_i\|\mu-\mu^*\|_2^2 = \sum_{i\in S_{\text{good}}} \langle X_i-\mu^*,\mu-\mu^*\rangle + \sum_{i\in S_{\text{good}}} (w_i-1)\langle X_i-\mu^*,\mu-\mu^*\rangle$$
$$- \sum_{i\in S_{\text{bad}}} \langle Y_i-\mu^*,\mu-\mu^*\rangle + \sum_{i\in S_{\text{bad}}} w_i\langle X_i-\mu,\mu-\mu^*\rangle \,.$$

Now we use $\vdash_t \ (x+y+z+w)^t \le \exp(t)\cdot(x^t+y^t+z^t+w^t)$ for any even t, and Lemma 4.6.6, Lemma 4.6.7, and Lemma 4.6.9 and simplify to conclude

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i\in S_{\text{good}}} w_i\right)^t \|\mu-\mu^*\|_2^{2t} \le \exp(t)\cdot t^{t/2}\cdot n^t\cdot\varepsilon^{t-1}\cdot\|\mu-\mu^*\|_2^t \,.$$

Lastly, since $\mathcal{A}\vdash_2 \sum_{i\in T} w_i \ge (1-2\varepsilon)n$, we get

$$\mathcal{A} \vdash_{O(t)} \|\mu-\mu^*\|_2^{2t} \le \exp(t)\cdot t^{t/2}\cdot\varepsilon^{t-1}\cdot\|\mu-\mu^*\|_2^t \,,$$

as claimed. □

4.6.5 Rounding

The rounding phase of our algorithm is extremely simple. If $\tilde{\mathbb{E}}$ satisfies $\mathcal{A}$, we have by Lemma 4.6.5 and pseudoexpectation Cauchy-Schwarz that

$$\tilde{\mathbb{E}}\,\|\mu-\mu^*\|_2^{2t} \le \exp(t)\cdot t^{t/2}\cdot\varepsilon^{t-1}\cdot\tilde{\mathbb{E}}\left(\|\mu-\mu^*\|_2^t\right) \le \exp(t)\cdot t^{t/2}\cdot\varepsilon^{t-1}\cdot\tilde{\mathbb{E}}\left(\|\mu-\mu^*\|_2^{2t}\right)^{1/2}$$

which implies that

$$\tilde{\mathbb{E}}\,\|\mu-\mu^*\|_2^{2t} \le \exp(t)\cdot t^t\cdot\varepsilon^{2(t-1)} \,. \tag{4.5}$$

Once this is known, analyzing $\| \tilde{\mathbb{E}} \mu - \mu^* \|_2$ is straightforward. By (4.5) and pseudo-Cauchy-Schwarz again,

$$\| \tilde{\mathbb{E}}[\mu] - \mu^* \|_2^2 \leq \tilde{\mathbb{E}} \| \mu - \mu^* \|_2^2 \leq \left(\tilde{\mathbb{E}} \| \mu - \mu^* \|_2^{2t} \right)^{1/t} \leq O(t \cdot \varepsilon^{2-2/t}) ,$$

which finishes analyzing the algorithm.

4.6.6 Proofs of Lemmata 4.6.6–4.6.9

We first prove Lemma 4.6.6, which is a relatively straightforward application of SoS Cauchy Schwarz.

Proof of Lemma 4.6.6. We have

$$\begin{aligned} \vdash_{O(t)} \left(\sum_{i \in S_{\text{good}}} \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t &= \left(\left\langle \sum_{i \in S_{\text{good}}} (X_i - \mu^*), \mu - \mu^* \right\rangle \right)^t \\ &\leq \left\| \sum_{i \in S_{\text{good}}} (X_i - \mu^*) \right\|_2^t \| \mu - \mu^* \|_2^t \\ &\leq \left(n \cdot O\left(\varepsilon^{1-1/t} \right) \cdot t^{1/2} \right)^t \| \mu - \mu^* \|_2^t , \end{aligned}$$

where the last inequality follows from (E3). This completes the proof. □

Before we prove Lemmata 4.6.7–4.6.9, we prove the following lemma which we will use repeatedly:

Lemma 4.6.10. *Let ε, t, μ^* and $Y_1, \ldots, Y_n \in \mathbb{R}^d$ be as in Theorem 4.6.1, and suppose they satisfy (E4). Then, we have*

$$\mathcal{A} \vdash_{O(t)} \sum_{i \in [n]} \langle Y_i - \mu^*, \mu - \mu^* \rangle^t \leq 2nt^{t/2} \| \mu - \mu^* \|_2^t .$$

Proof. We have that

$$\begin{aligned}
\vdash_t \sum_{i\in[n]} \langle Y_i - \mu^*, \mu - \mu^*\rangle^t &= \left[(\mu-\mu^*)^{\otimes 2}\right]^\top \sum_{i\in[n]} \left[(Y_i-\mu^*)^{\otimes t/2}\right] \left[(Y_i-\mu^*)^{\otimes t/2}\right]^\top \left[(\mu-\mu^*)^{\otimes 2}\right] \\
&\overset{(a)}{\leq} n\left(\left[(\mu-\mu^*)^{\otimes 2}\right]^\top \left(\mathop{\mathbb{E}}_{X\sim D}\left[(X-\mu^*)^{\otimes t/2}\right]\left[(X-\mu^*)^{\otimes t/2}\right]^\top + 0.1\cdot I\right)\left[(\mu-\mu^*)^{\otimes 2}\right]\right) \\
&= n\cdot \mathop{\mathbb{E}}_{X\sim D}\langle X-\mu^*, \mu-\mu^*\rangle^t + n\cdot 0.1\cdot \|\mu-\mu^*\|_2^t \\
&\overset{(b)}{\leq} 2n\cdot t^{t/2}\|\mu-\mu^*\|_2^t\,,
\end{aligned}$$

where (a) follows from (E4) and (b) follows from $10t$-explicitly boundedness. □

We now return to the proof of the remaining Lemmata.

Proof of Lemma 4.6.7. We start by applying Hölder's inequality, Fact D.1.6, (implicitly using that $w_i^2 = w_i \vdash_2 (1-w_i)^2 = 1-w_i$), to get

$$\begin{aligned}
\mathcal{A}\vdash_{O(t)} \left(\sum_{i\in S_{\text{good}}} (w_i - 1)\langle X_i-\mu^*, \mu-\mu^*\rangle\right)^t &= \left(\sum_{i\in S_{\text{good}}} (1-w_i)\langle X_i-\mu^*, \mu-\mu^*\rangle\right)^t \\
&\leq \left(\sum_{i\in S_{\text{good}}} (w_i-1)\right)^{t-1}\left(\sum_{i\in S_{\text{good}}} \langle X_i-\mu^*, \mu-\mu^*\rangle^t\right)\,.
\end{aligned}$$

By Lemma 4.6.10, we have

$$\begin{aligned}
\mathcal{A}\vdash_{O(t)} \sum_{i\in S_{\text{good}}} \langle X_i-\mu^*, \mu-\mu^*\rangle^t &\leq \sum_{i\in[n]} \langle Y_i-\mu^*, \mu-\mu^*\rangle^t \\
&\leq 2n\cdot t^{t/2}\cdot \|\mu-\mu^*\|_2^t\,.
\end{aligned}$$

At the same time,

$$A\vdash_2 \sum_{i\in T}(1-w_i) = (1-\varepsilon)n - \sum_{i\in[n]} w_i + \sum_{i\notin T} w_i = \sum_{i\notin T} w_i \leq \varepsilon n\,.$$

So putting it together, we have

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in T} (w_i - 1) \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t \leq 2(\varepsilon n)^{t-1} \cdot n \cdot t^{t/2} \cdot \|\mu - \mu^*\|_2^t \ ,$$

as claimed. □

Proof of Lemma 4.6.8. We apply Hölder's inequality to obtain that

$$\begin{aligned} \vdash_{O(t)} \left(\sum_{i \in S_{\mathrm{bad}}} \langle X_i - \mu^*, \mu - \mu^* \rangle \right)^t &\leq |S_{\mathrm{bad}}|^{t-1} \sum_{i \in S_{\mathrm{bad}}} \langle Y_i - \mu^*, \mu - \mu^* \rangle^t \\ &\overset{(a)}{\leq} (\varepsilon n)^{t-1} \sum_{i \in [n]} \langle Y_i - \mu^*, \mu - \mu^* \rangle^t \\ &\overset{(b)}{\leq} 2(\varepsilon n)^{t-1} n t^{t/2} \|\mu - \mu^*\|_2^t \ , \end{aligned}$$

where (a) follows from the assumption on the size of S_{bad} and since the additional terms in the sum are SoS, and (b) follows follows from Lemma 4.6.10. This completes the proof. □

Proof of Lemma 4.6.9. The proof is very similar to the proof of the two previous lemmas, except that we use the moment bound inequality in $\mathcal{A}$. Getting started, by Hölder's:

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_{\mathrm{bad}}} w_i \langle X_i - \mu, \mu - \mu^* \rangle \right)^t \leq \left(\sum_{i \in S_{\mathrm{bad}}} w_i \right)^{t-1} \left(\sum_{i \in S_{\mathrm{bad}}} w_i \langle X_i - \mu, \mu - \mu^* \rangle^t \right)$$

By evenness of t,

$$\vdash_t \sum_{i \in S_{\mathrm{bad}}} w_i \langle X_i - \mu, \mu - \mu^* \rangle^t \leq \sum_{i \in [n]} w_i \langle X_i - \mu, \mu - \mu^* \rangle^t \ .$$

Combining this with the moment bound in $\mathcal{A}$,

$$\mathcal{A} \vdash_{O(t)} \left(\sum_{i \in S_{\mathrm{bad}}} w_i \langle X_i - \mu, \mu - \mu^* \rangle \right)^t \leq \left(\sum_{i \in S_{\mathrm{bad}}} w_i \right)^{t-1} \cdot 2 \cdot t^{t/2} \cdot n \cdot \|\mu - \mu^*\|_2^t \ .$$

Finally, clearly $\mathcal{A} \vdash_2 \sum_{i \notin T} w_i \leq \varepsilon n$, which finishes the proof. $\square$

4.7 Encoding structured subset recovery with polynomials

The goal in this section is to prove Lemma 4.4.1. The eventual system $\widehat{\mathcal{A}}$ of polynomial inequalities we describe will involve inequalities among matrix-valued polynomials. We start by justifying the use of such inequalities in the SoS proof system.

4.7.1 Matrix SoS proofs

Let $x = (x_1, \ldots, x_n)$ be indeterminates. We describe a proof system which can reason about inequalities of the form $M(x) \succeq 0$, where $M(x)$ is a symmetric matrix whose entries are polynomials in x.

Let $M_1(x), \ldots, M_m(x)$ be symmetric matrix-valued polynomials of x, with $M_i(x) \in \mathbb{R}^{s_i \times s_i}$, and let $q_1(x), \ldots, q_m(x)$ be scalar polynomials. (If $s_i = 1$ then M_i is a scalar valued polynomial.) Let $M(x)$ be another matrix-valued polynomial. We write

$$\{M_1 \succeq 0, \ldots, M_m \succeq 0, q_1(x) = 0, \ldots, q_m(x) = 0\} \vdash_d M \succeq 0$$

if there are vector-valued polynomials $\{r_S^j\}_{j \leq N, S \subseteq [m]}$ (where the S's are multisets), a matrix B, and a matrix Q whose entries are polynomials in the ideal generated by $q_1, \ldots, q_m$, such that

$$M = B^\top \left[\sum_{S \subseteq [m]} \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \right] B + Q(x)$$

and furthermore that $\deg \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \leq d$ for every $S \subseteq [m]$, and $\deg Q \leq d$. Observe that in the case $M_1, \ldots, M_m, M$ are actually 1×1 matrices, this reduces to the usual notion of scalar-valued sum of squares proofs.

Adapting pseudodistributions to the matrix case, we say a pseudodistribution $\tilde{\mathbb{E}}$ of degree $2d$ satisfies the inequalities $\{M_1(x) \succeq 0, \ldots, M_m(x) \succeq 0\}$ if for every multiset $S \subseteq [m]$ and $p \in \mathbb{R}[x]$ such that $\deg \left[p(x)^2 \cdot (\otimes_{i \in S} M_i(x))\right] \leq 2d$,

$$\tilde{\mathbb{E}} \left[p(x)^2 \cdot (\otimes_{i \in S} M_i(x))\right] \succeq 0 \,.$$

For completeness, we prove the following lemmas in the appendix.

Lemma 4.7.1 (Soundness). *Suppose $\tilde{\mathbb{E}}$ is a degree-$2d$ pseudodistribution which satisfies constraints $\{M_1 \succeq 0, \ldots, M_m \succeq 0\}$, and*

$$\{M_1 \succeq 0, \ldots, M_m \succeq 0\} \vdash_{2d} M \succeq 0 \,.$$

Then $\tilde{\mathbb{E}}$ satisfies $\{M_1 \succeq 0, \ldots, M_m \succeq 0, M \succeq 0\}$.

Lemma 4.7.2. *Let $f(x)$ be a degree-ℓ s-vector-valued polynomial in indeterminates x. Let $M(x)$ be a $s \times s$ matrix-valued polynomial of degree ℓ'. Then*

$$\{M \succeq 0\} \vdash_{\ell\ell'} \langle f(x), M(x) f(x) \rangle \geq 0 \,.$$

Polynomial-time algorithms to find pseudodistributions satisfying matrix-SoS constraints follow similar ideas as in the non-matrix case. In particular, recall that to enforce a scalar constraint $\{p(x) \geq 0\}$, one imposes the convex constraint $\tilde{\mathbb{E}}\, p(x)(x^{\otimes d})(x^{\otimes d})^\top \succeq 0$. Enforcing a constraint $\{M(x) \succeq 0\}$ can be accomplished similarly by adding constraints of the form $\tilde{\mathbb{E}}\, M(x) \succeq 0, \tilde{\mathbb{E}}\, M(x) p(x) \succeq 0$, etc.

4.7.2 Warmup: Gaussian moment matrix-polynomials

In this section we develop the encoding as low degree polynomials of the following properties of an n-variate vector w and a d-variate vector μ. We will not be able to encode exactly these properties, but they will be our starting point. Let $d, n \in \mathbb{N}$, and suppose there are some vectors (a.k.a. samples) $X_1, \ldots, X_n \in \mathbb{R}^d$.

1. Boolean: $w \in \{0,1\}^n$.

2. Size: $(1-\tau)\alpha n \leq \sum_{i\in[n]} w_i \leq (1+\tau)\alpha n$.

3. Empirical mean: $\mu = \frac{1}{\sum_{i\in[n]} w_i} \sum_{i\in[n]} w_i X_i$.

4. t-th Moments: the t-th empirical moments of the vectors selected by the vector w, centered about μ, are subgaussian. That is,

$$\max_{u\in\mathbb{R}^d} \frac{1}{\alpha n} \sum_{i\in[n]} w_i \langle X_i - \mu, u\rangle^t \leq 2 \cdot t^{t/2} \|u\|_2^t \,.$$

The second property is already phrased as two polynomial inequalities, and the third can be rearranged to a polynomial equation. For the first, we use polynomial equations $w_i^2 = w_i$ for every $i \in [n]$. The moment constraint will be the most difficult to encode. We give two versions of this encoding: a simple one which will work when the distribution of the structured subset of samples to be recovered is Gaussian, and a more complex version which allows for any explicitly bounded distribution. For now we describe only the Gaussian version. We state some key lemmas and prove them for the Gaussian case. We carry out the general case in the following section.

To encode the bounded moment constraint, for this section we let $M(w,\mu)$ be the following matrix-valued polynomial

$$M(w,\mu) = \frac{1}{\alpha n} \sum_{i\in[n]} w_i \left[(X_i - \mu)^{\otimes t/2}\right] \left[(X_i - \mu)^{\otimes t/2}\right]^\top$$

Definition 4.7.1 (Structured subset axioms, Gaussian version)**.** For parameters $\alpha \in [0,1]$ (for the size of the subset), t (for which empirical moment to control), and $\tau > 0$ (to account for some empirical deviations), the structured subset axioms are the following matrix-polynomial inequalities on variables $w = (w_1, \ldots, w_n), \mu = (\mu_1, \ldots, \mu_d)$.

1. booleanness: $w_i^2 = w_i$ for all $i \in [n]$

2. size: $(1-\tau)\alpha n \leq \sum_{i\in[n]} w_i \leq (1+\tau)\alpha n$

3. t-th moment boundedness: $M(w,\mu) \preceq 2 \cdot \mathbb{E}_{X\sim\mathcal{N}(0,I)} \left[X^{\otimes t/2}\right] \left[X^{\otimes t/2}\right]^\top$.

4. μ is the empirical mean: $\mu \cdot \sum_{i \in [n]} w_i = \sum_{i \in [n]} w_i X_i$.

Notice that in light of the last constraint, values for the variables μ are always determined by values for the variables w, so strictly speaking μ could be removed from the program. However, we find it notationally convenient to use μ. We note also that the final constraint, that μ is the empirical mean, will be used only for the robust statistics setting but seems unnecessary in the mixture model setting.

Next, we state and prove some key lemmas for this Gaussian setting, as warmups for the general setting.

Lemma 4.7.3 (Satisfiability, Gaussian case)**.** *Let $d \in \mathbb{N}$ and $\alpha = \alpha(d) > 0$. Let $t \in \mathbb{N}$. Suppose $(1-\tau)\alpha n \geq d^{100t}$. Let $0.1 > \tau > 0$. If $X_1, \ldots, X_n \in \mathbb{R}^d$ has a subset $S \subseteq [n]$ such that $\{X_i\}_{i \in S}$ are iid samples from $\mathcal{N}(\mu^*, I)$ and $|S| \geq (1-\tau)\alpha n$, then with probability at least $1 - d^{-8}$ over these samples, the α, t, τ structured subset axioms are satisfiable.*

Proof. Suppose S has size exactly $(1-\tau)\alpha n$; otherwise replace S with a random subset of S of size exactly $(\alpha - \tau)n$. As a solution to the polynomials, we will take w to be the indicator vector of S and $\mu = \frac{1}{|S|}\sum_{i \in [n]} w_i X_i$. The booleanness and size axioms are trivially satisfied. The spectral inequality

$$\frac{1}{\alpha n} \sum_{i \leq [n]} w_i \left[(X_i - \mu)^{\otimes t/2}\right] \left[(X_i - \mu)^{\otimes t/2}\right]^\top \preceq 2 \cdot \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} \left[X^{\otimes t/2}\right] \left[X^{\otimes t/2}\right]^\top$$

follows from concentration of the empirical mean to the true mean μ^* and standard matrix concentration (see e.g. [Tro12]). □

The next lemma is actually a corollary of Lemma 4.7.2.

Lemma 4.7.4 (Moment bounds for polynomials of μ, Gaussian case)**.** *Let $f(\mu)$ be a length-d vector of degree-ℓ polynomials in indeterminates $\mu = (\mu_1, \ldots, \mu_k)$. The t-th moment boundedness axiom implies the following inequality with a degree $t\ell$ SoS* *proof.*

$$\left\{ M(w,\mu) \preceq 2 \cdot \mathop{\mathbb{E}}_{X\sim\mathcal{N}(0,I)} \left[X^{\otimes t/2}\right]\left[X^{\otimes t/2}\right]^\top \right\}$$
$$\vdash_{O(t\ell)} \frac{1}{\alpha n}\sum_{i\in[n]} w_i \langle X_i-\mu, f(\mu)\rangle^t \leq 2 \cdot \mathop{\mathbb{E}}_{X\sim\mathcal{N}(0,I)} \langle X, f(\mu)\rangle^t \,.$$

4.7.3 Moment polynomials for general distributions

In this section we prove Lemma 4.4.1.

We start by defining polynomial equations $\widehat{\mathcal{A}}$, for which we introduce some extra variables For every pair of multi-indices γ, ρ over $[k]$ with degree at most $t/2$, we introduce a variable $M_{\gamma,\rho}$. The idea is that $M = [M_{\gamma,\rho}]_{\gamma,\rho}$ forms an $n^{t/2} \times n^{t/2}$ matrix. By imposing equations of the form $M_{\gamma,\rho} = f_{\gamma,\rho}(w,\mu)$ for some explicit polynomials $f_{\gamma,\rho}$ of degree $O(t)$, we can ensure that

$$\langle u^{\otimes t/2}, M u^{\otimes t/2}\rangle = 2\cdot t^{t/2}\|u\|_2^t - \frac{1}{\alpha n}\sum_{i\in[n]} w_i\langle X_i - \mu, u\rangle^t \,.$$

(This equation should be interpreted as an equality of polynomials in indeterminates u.) Let $\mathcal{L}$ be such a family of polynomial equations. Our final system $\widehat{\mathcal{A}}(\alpha,t,\tau)$ of polynomial equations and inequalities follows. The important parameters are α, controlling the size of the set of samples to be selected, and t, how many moments to control. The parameter τ is present to account for random fluctuations in the sizes of the cluster one wants to recover.

Definition 4.7.2. Let $\widehat{\mathcal{A}}(\alpha,t,\tau)$ be the set of (matrix)-polynomial equations and inequalities on variables $w, \mu, M_{\gamma,\rho}$ containing the following.

1. Booleanness: $w_i^2 = w_i$ for all $i \in [n]$

2. Size: $(1-\tau)\alpha n \leq \sum w_i \leq (1+\tau)\alpha n$.

3. Empirical mean: $\mu \cdot \sum_{i\in[n]} w_i = \sum_{i\in[n]} w_i X_i$.

4. The equations $\mathcal{L}$ on M described above.

5. Positivity: $M \succeq 0$.

In the remainder of this section we prove the satisfiability and moment bounds parts of Lemma 4.4.1. To prove the lemma we will need a couple of simple facts about SoS proofs.

Fact 4.7.5. *Let $X_1, \ldots, X_m \in \mathbb{R}^d$. Let $v \in \mathbb{R}^d$ have $\|v\|_2 \leq 1$. Let $Y_i = X_i + v$. Let $t \in \mathbb{N}$ be even. Suppose there is $C \in \mathbb{R}$ with $C \geq 1$ such that for all $s \leq t$,*

$$\frac{1}{m} \sum_{i \in [m]} \|X_i\|_2^s \leq C^s$$

Then

$$\vdash_t \frac{1}{m} \sum_{i \in [n]} \left[\langle X_i, u \rangle^t - \langle Y_i, u \rangle^t \right] \leq \left(2^t C^{t-1} \|v\|_2 \right) \|u\|_2^t$$

and similarly for $\frac{1}{m} \sum_{i \in [n]} [\langle Y_i, u \rangle^t - \langle X_i, u \rangle^t]$.

Proof. Expanding $\langle Y_i, u \rangle^t$, we get

$$\langle Y_i, u \rangle^t = \langle X_i + v, u \rangle^t = \sum_{s \leq t} \binom{t}{s} \langle X_i, v \rangle^s \langle v, u \rangle^{t-s} .$$

So,

$$\frac{1}{m} \sum_{i \in [m]} \left[\langle X_i, u \rangle^t - \langle Y_i, u \rangle^t \right] = -\frac{1}{m} \sum_{i \in [m]} \sum_{s < t} \binom{t}{s} \langle X_i, u \rangle^s \langle v, u \rangle^{t-s} .$$

For each term, by Cauchy-Schwarz, $\vdash_t \langle X_i, u \rangle^s \langle v, u \rangle^{t-s} \leq \|X_i\|_2^s \|v\|_2^{t-s} \cdot \|u\|_2^t$. Putting these together with the hypothesis on $\frac{1}{n} \|X_i\|_2^s$ and counting terms finishes the proof. □

Proof of Lemma 4.4.1: Satisfiability. By taking a random subset S if necessary, we assume $|S| = (1-\tau)\alpha n = m$. We describe a solution to the system $\widehat{\mathcal{A}}$. Let w be the 0/1 indicator vector for S. Let $\mu = \frac{1}{m} \sum_{i \in S} w_i X_i$. This satisfies the Boolean-ness, size, and empirical mean axioms.

Describing the assignment to the variables $\{M_{\gamma,\rho}\}$ takes a little more work. Re-indexing and centering, let $Y_1 = X_{i_1} - \mu, \ldots, Y_m = X_{i_m} - \mu$ be centered versions of the samples in S, where $S = \{i_1, \ldots, i_m\}$ and μ remains the empirical mean $\frac{1}{m}\sum_{i \in S} X_i$. First suppose that the following SoS proof exists:

$$\vdash_t \frac{1}{\alpha n} \sum_{i \in S} \langle Y_i, u \rangle^t \leq 2 \cdot t^{t/2} \|u\|_2^t \, .$$

Just substituting definitions, we also obtain

$$\vdash_t \frac{1}{\alpha n} \sum_{i \in [n]} w_i \langle X_i - \mu, u \rangle^t \leq 2 \cdot t^{t/2} \|u\|_2^t \, .$$

where now w and μ are scalars, not variables, and u are the only variables remaining. The existence of this SoS proof means there is a matrix $P \in \mathbb{R}^{d^{t/2} \times d^{t/2}}$ such that $P \succeq 0$ and

$$\langle u^{\otimes t/2}, P u^{\otimes t/2} \rangle = 2 t^{t/2} \|u\|_2^t - \frac{1}{\alpha n} \sum_{i \in [n]} w_i \langle X_i - \mu, u \rangle^t \, .$$

Let $M_{\gamma,\rho} = P_{\gamma,\rho}$. Then clearly $M \succeq 0$ and M, w, μ together satisfy $\mathcal{L}$.

It remains to show that the first SoS proof exists with high probability for large enough m. Since t is even and $0.1 > \tau > 0$, it is enough to show that

$$\vdash_t \frac{1}{m} \sum_{i \in [S]} \langle Y_i, u \rangle^t \leq 1.5 \cdot t^{t/2} \|u\|_2^t$$

Let $Z_i = X_i - \mu^*$, where μ^* is the true mean of D. Let

$$a(u) = \frac{1}{m} \sum_{i \in S} \left[\langle Z_i, u \rangle^t - \langle Y_i, u \rangle^t \right] \qquad b(u) = \frac{1}{m} \sum_{i \in S} \langle Z_i, u \rangle^t - \mathop{\mathbb{E}}_{Z \sim D - \mu^*} \langle Z, u \rangle^t \, .$$

We show that for $d \geq 2$,

$$\vdash_t a(u) \leq \tfrac{1}{4} \|u\|_2^t \qquad \vdash_t b(u) \leq \tfrac{1}{4} \|u\|_2^t$$

so long as the following hold

1. (bounded norms) for every $s \leq t$ it holds that $\frac{1}{m}\sum_{i \in [m]} \|Z_i\|_2^s \leq s^{100s} d^{s/2}$.

2. (concentration of empirical mean) $\|\mu - \mu^*\|_2 \leq d^{-5t}$.

3. (bounded coefficients) For every multiindex θ of degree $|\theta| = t$, one has

$$\left| \frac{1}{m} \sum_{i \in [m]} Z_i^\theta - \mathop{\mathbb{E}}_{Z \sim D} Z^\theta \right| \leq d^{-10t} \, .$$

We verify in Fact 4.7.6 following this proof that these hold with high probability by standard concentration of measure, for $m \geq d^{100t}$ and D $10t$-explicitly bounded, as assumed. Together with the assumption $\vdash_t \mathbb{E}_{Z \sim D - \mu^*} \langle Z, u \rangle^t \leq t^{t/2} \|u\|_2^t$, this will conclude the proof.

Starting with $a(u)$, using Fact 4.7.5, it is enough that $2^t C^{t-1} \|v\|_2 \leq \frac{1}{4}$, where $v = \mu - \mu^*$ and C is such that $\frac{1}{m}\sum_{i \in [m]} \|Z_i\|_2^s \leq C^s$. By 1 and 2, we can assume $\|v\|_2 \leq d^{-5t}$ and $C = t^{100} d^{1/2}$. Then the conclusion follows for $t \geq 3$.

We turn to $b(u)$. A typical coefficient of $b(u)$ in the monomial basis—say, the coefficient of u^θ for some multiindiex θ of degree $|\theta| = t$, looks like

$$\frac{1}{m} \sum_{i \in [m]} Y_i^\theta - \mathop{\mathbb{E}}_{Y \sim D} Y^\theta \, .$$

By assumption this is at most d^{-10t} in magnitude, so the sum of squared coefficients of $b(u)$ is at most d^{-18t}. The bound on $b(u)$ for $d \geq 2$. □

Proof of Lemma 4.4.1: Moment bounds. As in the lemma statement, let $f(\mu)$ be a vector of degree-ℓ polynomials in μ. By positivity and Lemma 4.7.2,

$$M(w, \mu) \succeq 0 \vdash_{O(t\ell)} \langle f(\mu)^{\otimes t/2}, M(w, \mu) f(\mu)^{\otimes t/2} \rangle \geq 0 \, .$$

Using this in conjunction with the linear equations $\mathcal{L}$,

$$\widehat{\mathcal{A}} \vdash_{O(t\ell)} 2t^{t/2}\|f(\mu)\|_2^t - \frac{1}{\alpha n}\sum_{i\in[n]} w_i\langle X_i - \mu, f(\mu)\rangle^t \geq 0$$

which is what we wanted to show. □

Fact 4.7.6 (Concentration for items 1, 2,3)**.** *Let $d, t \in \mathbb{N}$. Let D be a mean-zero distribution on $\mathbb{R}^d$ such that $\mathbb{E}\langle Z, u\rangle^s \leq s^s\|u\|_2^s$ for all $s \leq 10t$ for every $u \in \mathbb{R}^d$. Then for $t \geq 4$ and large enough d and $m \geq d^{100t}$, for m independent samples $Z_1, \ldots, Z_m \sim D$,*

1. *(bounded norms) for every $s \leq t$ it holds that $\frac{1}{m}\sum_{i\in[m]}\|Z_i\|_2^s \leq s^{100s}d^{s/2}$.*
2. *(concentration of empirical mean) $\left\|\frac{1}{m}\sum_{i\in[m]} Z_i\right\| \leq d^{-5t}$.*
3. *(bounded coefficients) For every multiindex θ of degree $|\theta| = t$, one has*

$$\left|\tfrac{1}{m}\sum_{i\in[m]} Z_i^\theta - \mathop{\mathbb{E}}_{Z\sim D} Z^\theta\right| \leq d^{-10t} .$$

Proof. The proofs are standard applications of central limit theorems, in particular the Berry-Esseen central limit theorem [Ber41], since all the quantities in question are sums of iid random variables with bounded moments. We will prove only the first statement; the others are similar.

Note that $\frac{1}{m}\sum_{i\in[m]}\|Z_i\|_2^s$ is a sum of iid random variables. Furthermore, by our moment bound assumption, $\mathbb{E}_{Z\sim D}\|Z\|_2^s \leq s^{2s}d^{s/2}$. We will apply the Berry-Esseen central limit theorem [Ber41]. The second and third moments $\mathbb{E}(\|Z\|_2^s - \mathbb{E}\|Z\|_2^s)^2, \mathbb{E}(\|Z\|_2^s - \mathbb{E}\|Z\|_2^s)^3$ are bounded, respectively, as $s^{O(s)}k^s$ and $s^{O(s)}d^{3s/2}$. By Berry-Esseen,

$$\Pr\left\{\frac{\sqrt{m}}{d^{s/2}} \cdot \frac{1}{m}\sum_{i\in[m]}\|Z_i\|_2^s > r + \frac{\sqrt{m}}{d^{s/2}}\mathbb{E}\|Z\|_2^s\right\} \leq e^{-\Omega(r^2)} + s^{O(s)} \cdot m^{-1/2} .$$

□

Finally we remark on the polynomial-time algorithm to find a pseudoexpectation satisfying $\widehat{\mathcal{A}}$. As per [BS17], it is just necessary to ensure that if $x = (w, \mu)$, the polynomials in $\widehat{\mathcal{A}}$ include $\|x\|_2^2 \leq M$ for some large number M. In our case the equation $\|x\|_2^2 \leq (nkm)^{O(1)}$ can be added without changing any arguments.

4.7.4 Modifications for robust estimation

We briefly sketch how the proof of Lemma 4.4.1 may be modified to prove Lemma 4.6.3. The main issue is that $\widehat{\mathcal{A}}$ of Lemma 4.4.1 is satisfiable when there exists an SoS proof

$$\vdash_t \frac{1}{(1-\varepsilon)n} \sum_{i \in [n]} w_i \langle X_i - \mu, u \rangle^t \leq 2t^{t/2} \|u\|_2^t$$

where μ is the empirical mean of X_i such that $w_i = 1$. In the proof of Lemma 4.4.1 we argued that this holds when w is the indicator for a set of iid samples from a $10t$-explicitly bounded distribution D. However, in the robust setting, w should be taken to be the indicator of the $(1-\varepsilon)n$ good samples remaining from such a set of iid samples after εn samples are removed by the adversary. If $Y_1, \ldots, Y_n$ are the original samples, with empirical mean μ^*, the proof of Lemma 4.4.1 (with minor modifications in constants) says that with high probability,

$$\vdash_t \frac{1}{n} \sum_{i \in [n]} \langle Y_i - \mu^*, u \rangle^t \leq 1.1 t^{t/2} \|u\|_2^t$$

For small-enough ε, this also means that

$$\vdash_t \frac{1}{(1-\varepsilon)n} \sum_{i \text{ good}} \langle X_i - \mu^*, u \rangle^t \leq 1.2 t^{t/2} \|u\|_2^t \,.$$

This almost implies that $\widehat{\mathcal{A}}$ is satisfiable given the ε-corrupted vectors $X_1, \ldots, X_n$ and parameter $\alpha = (1-\varepsilon)n$, except for that $\mu^* = \frac{1}{n} \sum_{i \in [n]} Y_i$ and we would like to replace it with $\mu = \frac{1}{(1-\varepsilon)n} \sum_{i \text{ good}} X_i$. This can be accomplished by noting that, as argued in Section 4.6, with high probability $\|\mu - \mu^*\|_2 \leq O(t \cdot \varepsilon^{1-1/t})$.

Chapter 5

Filtering I: Learning a High Dimensional Gaussian (and Beyond)

You're different; it's strange.
Pause to look at the change.
And though it's familiar
Still somehow I know
That from here a different story unfolds.

After the wild journey into madness that was the last couple of chapters, it is now time to take a step back, and go back to fundamentals. In this chapter we return to Problem 1.4.1, and present a different algorithm for this problem.

Rather than assign weights to individual points corresponding to our belief as to whether or not the point is corrupted or not, this framework will simply repeatedly throw away the points which it considers the most suspicious. The key point to our analysis will be to show that under a fixed set of determinstic conditions, the algorithm always (or in some cases, in expectation) throws away more corrupted points than uncorrupted points.

How does the algorithm decide how "suspicious" a point is? Recall the idea of spectral signatures, which were also key for the framework based on convex programming. For concreteness, consider the problem of robustly learning the mean of a

Gaussian (Problem 1.4.2). Previously, we showed that the top eigenvector of the covariance gave us a way to construct a separation oracle for the set of feasible weights. Intuitively, this is because on average the corrupted points should be further away in this direction. In this chapter, we will take this even further: we will show that under somewhat stronger concentration conditions, if we simply project all the data points onto the top eigenvalue, we can simply throw away the data points which are farthest away along this projection, and repeat this process until the spectral signature disappears.

This is a key insight in the design of filtering: these spectral signatures induce very simple ways of detecting which outliers are affecting the statistic at hand. As a result, we can use these very simple iterative procedures to reliably remove them.

The main advantage of this approach is that it is extremely efficient: a single iteration of filtering requires only (1) finding an approximate top eigenvector and eigenvalue of the covariance of the data, (2) checking if this eigenvalue is above a certain threshold, and if it is, (3) projecting all the data points on the eigenvector, and throwing away the largest. All these steps can be done in nearly linear time, and therefore in most cases, a single iteration of filtering runs in nearly linear time. As we shall see, we can show that in many cases, filtering is guaranteed to finish in very few iterations. As a result, the overall algorithm has very good runtime guarantees in theory. In fact, in practice we found that the algorithm does even better: often 3-4 iterations suffice to remove almost all outliers.

The downside is that because we are somewhat more careless with individual data points, this algorithm requires somewhat stronger concentration conditions on the uncorrupted data points. However, we are able to show that this price is not too high—indeed, in many settings we pay only a polylogarithmic overhead.

5.1 Additional preliminaries

In this chapter and going forward, it will be useful to have notation to deal with empirical means and covariances of data sets. This is because our arguments will (as opposed to before) very explicitly change the "active" data set in every iteration by removing points from it.

Let $S \subseteq \mathbb{R}^d$ be any finite set. We let $X \in_u S$ denote a uniformly random draw over S. We will let

$$\mu^S = \mathop{\mathbb{E}}_{X \in_u S}[X] = \frac{1}{|S|} \sum_{X \in S} X \qquad (5.1)$$

denote the empirical mean, and for any $y \in \mathbb{R}^d$ we let

$$M^S(y) = \mathop{\mathbb{E}}_{X \in_u S}\left[(X - y)(X - y)^\top\right] = \frac{1}{|S|} \sum_{X \in S} (X - y)(X - y)^\top \qquad (5.2)$$

denote a modified version of the empirical covariance, which is equal to the covariance of the uniform distribution over S when $y = \mu^S$.

We also require the following definition that quantifies the extent to which a set of samples has been corrupted:

Definition 5.1.1. Given finite sets S and S' we let $\Delta(S, S') = \frac{|S \Delta S'|}{|S|}$ be the size of the symmetric difference of S and S' divided by the cardinality of S.

Finally, we require the following guarantee, which says that, given a matrix, it is possible to find an vector which captures a constant fraction of the energy of the top singular vector of the matrix, in nearly linear time. Formally:

Fact 5.1.1 ([MM15])**.** *Fix $\alpha, \delta > 0$. Let $A \in \mathbb{R}^{n \times d}$, and let*

$$\lambda^* = \sup_{\|u\|_2 = 1} u^\top A^\top A u \ ,$$

be the square of the top singular value of A. Then, there is an algorithm ApproxSVD(A, α, δ) *which runs in time $\widetilde{O}\left(nd \log \frac{1}{\alpha} \log \frac{1}{\delta}\right)$ which with probability $1 - \delta$ outputs a unit vector v so that*

$$v^\top A^\top A v \geq (1 - \alpha) \lambda^* \ .$$

We remark that more recently there have been more algorithms which achieve even faster runtimes (see e.g. [Sha16, AZL16]). However, in the regimes that we will care about (i.e. when $\alpha = \Omega(1)$), this more basic guarantee suffices.

For simplicity, we will assume that ApproxSVD always succeeds. Since its runtime dependence on δ, the failure probability, is logarithmic, it should be clear that since we will only call ApproxSVD polynomially many times (in fact, $\widetilde{O}(d \log 1/\varepsilon)$ times), by taking $\delta' = \mathrm{poly}(1/n, 1/d, \delta)$, we lose only logarithmic factors in the runtime and we may assume that all runs of ApproxSVD succeed.

5.2 General pseudocode for filtering

In this chapter (and going forward), it will be very useful to state the filtering framework in very general framework, as we will be instantiating it in a wide variety of settings. We present the general framework in 14. In addition to the dataset S which is to be filtered, filtering requires the following parameters and subroutines:

- ε, the fraction of points which are corrupted. We note that often the algorithm does not require this, or if it does, there are standard techniques to estimate ε.

- δ, the probability of failure we are willing to tolerate. This parameter is only really necessary for technical reasons and should be largely ignored.

- ComputeScores, a way to compute scores which are intended to measure how "suspicious" any individual data point is. We will think of this as a function $\tau : S \to \mathbb{R}$. In accordance to the discussion above regarding spectral signatures, in all instances this method will use some spectral method to determine the scores.

- Thres, a way to decide when to stop filtering. In theory, this should involve some check of whether or not the scores are too large or not in aggregate. However, in practice this often simply returns whether or not the algorithm has run for a fixed number of iterations.

- Remove, a way to remove data points based on the scores. Since scores are intended to be larger for more suspicious points, this will usually simply remove points which have large scores. However, the specific thresholds we choose will need to be problem dependent.

Algorithm 14 General filtering meta-algorithm

```
function GeneralFilter($S, \varepsilon, \delta$, ComputeScores, Thres, Remove)
    Let $n = |S|$
    Let $\tau \leftarrow$ ComputeScores($S$)
    if Thres($\tau, \varepsilon, \delta$) then
        return "DONE"
    else
        Let $S \leftarrow$ Remove($S, \tau, \varepsilon, \delta$)
```

Remark 5.2.1. For conciseness, when it is understood, we will often omit the parameters ε, δ from the list of inputs to GeneralFilter and its concrete instantiations.

Remark 5.2.2. We remark that there are a couple of cases where the algorithm doesn't technically fit this framework, or requires additional parameters. For instance, even the algorithm which simply runs the loop for a constant number of iterations technically cannot be described in this way without some additional state. However, we trust the reader can figure out how to implement these minor changes if necessary. For simplicity of presentation we will ignore these issues.

5.2.1 Spectral filtering

An important special case of this framework is the case where the scores are computed using approximate spectral methods. This will be what we use in almost every instance where we need to robustly estimate a mean. As we discuss above, this is because when the mean is corrupted, we should find a spectral signature. Therefore, data points which have large correlation with the top eigenvector of the empirical covariance should be considered suspicious, and thus it makes sense to assign them higher scores, in accordance to how much they contribute to the top eigenvector. The formal pseudocode is given in Algorithm 15.

Algorithm 15 Computing scores via a spectral criterion

```
function ComputeSpectralScores($S$)
    Let $n = |S|$
    Compute the sample mean $\mu^S$.
    Let $A$ be the matrix whose rows are given by $\frac{1}{\sqrt{n}}\left(X - \mu^S\right)$, for each $X \in S$.
    Let $\delta' = \mathrm{poly}(\varepsilon, \delta, 1/n, 1/d)$.
    Let $v = $ ApproxSVD$(A, 1/10, \delta')$.
    For $X \in S$, let $\tau(X) = (v^\top (X - \mu^S))^2$.
    return $\tau : S \to \mathbb{R}$
```

Note that if τ is the output of ComputeSpectralScores, then

$$\frac{1}{n}\sum_{X \in S} \tau(S) = \frac{1}{n}\sum_{X \in S} (v^\top (X - \mu^S))^2 = v^\top M^S(\mu^S) v \ , \tag{5.3}$$

where v is the approximate eigenvector found by ApproxSVD. Thus, these scores exactly correspond to how much each individual point contributes to the (approximate) top eigenvalue of the empirical covariance. With this algorithm, we also define the following important special case of GeneralFilter, which uses these spectral scores.

Definition 5.2.1. For any choice of Thres, Remove, we define

$$\textsc{SpectralFilter}(S, \varepsilon, \delta, \textsc{Thres}, \textsc{Remove}) = \textsc{GeneralFilter}(S, \varepsilon, \delta, \textsc{Thres}, \textsc{Remove}) \ .$$

We observe that since ApproxSVD runs in nearly linear time, it trivially follows that:

Corollary 5.2.1. *For any set of n points S in $\mathbb{R}^d$, we have that* ComputeSpectralScores(S) *runs in time $\widetilde{O}(nd)$.*

5.2.2 How do we choose the threshold and how to remove points?

If we use spectral criteria to determine the scores, how should use the scores to determine the threshold and the removal criteria? We give some high level intuition here, which will serve as a rough guideline for the algorithms we describe in detail below.

Removal criteria We will start with how to choose the removal criteria. To define this, one should first find the tightest univariate tail bound you can expect from your data set, if you had the true statistics. For instance, if your uncorrupted data is sub-Gaussian, then this should look like a sub-Gaussian style tail bound. If your uncorrupted data has bounded covariance, then this should be like a Chebyshev-style bound. Then, the removal algorithm should somehow attempt to remove points which cause the scores to violate this concentration bound. Unfortunately, the exact form of how this should be done seems to change depending on the form of the concentration bound. For instance, for the second moment method, we seem to inherently require randomness to get the right answer. We leave it as an interesting open question to give a simple, unified approach for designing the removal algorithm.

Threshold criteria The principles for designing the threshold algorithm are similar. We take the same univariate tail bound as before, and ask: given a distribution which satisfies this univariate tail bound, how much larger does the largest ε-fraction of scores make the overall mean of the scores? In general, we should set the threshold to be of this order, plus whatever the good points should contribute in expectation to this statistic. Roughly this is because this is the amount of deviation that the worst ε-fraction of points from the true distribution could contribute.

In the special case of spectral filtering we may derive a closed form formula for what the threshold ought to be in the infinite sample limit, given these considerations. In the examples we discuss below, this essentially gives the right answer. Let D be our distribution. In the infinite sample setting, we should think of τ as a function over all of $\mathbb{R}^d$. In this case, when we have the "right statistic", i.e., we have properly centered the distribution, so we may assume $\mathbb{E}_{X \sim D}[X] = 0$, the score function will exactly be $\tau(X) = (v^\top X)^2$, where v is the top eigenvector of the covariance Σ of the distribution.[1] Let Φ_v denote the CDF of D when projected onto v, and let ϕ_v denote

[1] We are also ignoring issues of approximation in this informal discussion; see Section 5.2.3 for a

the PDF. Then, the guideline says that the threshold should be

$$\mathfrak{T}_\varepsilon(D) = \mathop{\mathbb{E}}_{X \sim D_v}[X^2] + \int_{\Phi_v^{-1}(1-\varepsilon)}^{\infty} x^2 \phi_v(x) dx \; . \tag{5.4}$$

Notice that this should work for any $\tau(X) = f(v^\top X)$ where $f : \mathbb{R} \to \mathbb{R}$ is monotonic. In this case the expression is easily generalized:

$$\mathfrak{T}_\varepsilon(D, f) = \mathop{\mathbb{E}}_{X \sim D_v}[f(X)] + \int_{\Phi_v^{-1}(1-\varepsilon)}^{\infty} f(x) \phi_v(x) dx \; . \tag{5.5}$$

Of course, these are only general rules, and care must be applied to each individual situation to apply them. In fact, it doesn't even quite give the right answer for mean estimation under bounded second moment assumptions! However the interested reader may find them useful in trying to understand the algorithms given below.

5.2.3 Approximation, randomness, and other gremlins

Finally, we remark about a couple of minor points regarding approximation that we will ignore for the rest of the thesis.

For the rest of the thesis, we will typically assume that the vector that Algorithm 15 finds is exactly the top eigenvector of the empirical covariance. This is not true: ApproxSVD only guarantees that with high probability, we find a vector that has roughly the same energy as the top eigenvector of the empirical covariance. However, it is easy to verify that all of our arguments only require that the vector we find has roughly the same energy as the top eigenvector (except in one case, namely, in Section 5.3, but we will address the question explicitly there). There is also the issue that with some small probability, the algorithm fails, since ApproxSVD fails with some small probability. However, since the failure probability of ApproxSVD grows logarithmically with $1/\delta'$, by our choice of δ', the probability that any run fails is negligible so long as we only run ComputeSpectralScores only polynomially times, which we will always do. Thus for simplicity we will always assume that

discussion of issues of this sort

ApproxSVD always succeeds.

5.2.4 Organization

As in Chapter 2, it suffices to solve Problem 1.4.2 and Problem 1.4.3 separately. After giving algorithms for both, it is not hard to show (via very similar arguments as in Chapter 2) that this gives an algorithm for the full problem. Thus in this chapter we will only show how to solve the subproblems separately. In Section 5.3 we show how to use the filtering framework to learn the mean robustly, and in Section 5.4 we show how to learn the covariance robustly.

5.3 Learning the mean of an isotropic sub-Gaussian distribution

In this section, we use our filter technique to give an agnostic learning algorithm for learning the mean of an isotropic sub-Gaussian distribution with known covariance matrix. In particular observe this captures the case of isotropic Gaussians. More specifically, we prove:

Theorem 5.3.1. *Let $\varepsilon, \delta > 0$, and let $\mu \in \mathbb{R}^d$. Let $X_1, \ldots, X_n$ be an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size*

$$n = \Omega\left(\frac{d}{\varepsilon^2} \operatorname{poly}\log\left(\frac{d}{\varepsilon\delta}\right)\right) .$$

There exists an algorithm that, given $X_1, \ldots, X_n$ and $\varepsilon > 0$, returns a vector $\widehat{\mu}$ such that with probability at least $1 - \delta$ we have

$$\|\widehat{\mu} - \mu\|_2 = O(\varepsilon\sqrt{\log(1/\varepsilon)}) .$$

Moreover, the algorithm runs in time $\widetilde{O}(nd^2)$.

Observe that it requires $O(nd)$ time to read the samples, so this guarantees that the algorithm runs in essentially $O(d)$ reads of the input.

Throughout this section, we will let μ denote the (unknown) mean, and we let S' denote an ε-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size n. We will let S denote a set of i.i.d. samples from $\mathcal{N}(\mu, I)$ so that $\Delta(S, S') = \varepsilon$. By the definition of ε-corruption, we know that such an S exists.

Deterministic conditions As before, we start by defining our notion of good sample, i.e, a set of conditions on the uncorrupted set of samples under which our algorithm will succeed. These will correspond to the deterministic conditions defined in (2.5)-(2.7), but as we shall see, are somewhat more stringent.

Fix $\delta > 0$. We will prove that our algorithm will succeed under the following set of deterministic conditions on our data points S' and our uncorrupted data points S. We will require that (S, S') satisfy:

$$\Delta(S, S') = \varepsilon \tag{5.6}$$

$$\|X - \mu\|_2 \leq O(\sqrt{d \log(|S|/\delta)}) \text{ , } \forall X \in S \tag{5.7}$$

$$\|\mu^S - \mu\|_2 \leq \varepsilon \text{ ,} \tag{5.8}$$

$$\left\| M^S(\mu) - I \right\|_2 \leq \varepsilon \text{ .} \tag{5.9}$$

$$\left| \Pr_{X \in_u S}[L(X) \geq 0] - \Pr_{X \sim \mathcal{N}(\mu, I)}[L(X) \geq 0] \right| \leq \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\delta})\right)} \text{ ,} \tag{5.10}$$

$$\forall L : \mathbb{R}^d \to \mathbb{R} \text{ s.t. } L(x) = v \cdot (x - \mu) - T, \text{ where } T \geq 1,\ \|v\|_2 = 1$$

We pause briefly to interpret these conditions. Condition (5.6) is the only requirement on S', and it is the requirement that S' is close to S. Conditions (5.7)-(5.9) are completely standard and essentially state that no sample is too far away from the mean, and that the empirical mean and covariance converge.

The only really unusual condition is Condition (5.10). This condition is a statement about the convergence of linear threshold functions: it says that the fraction of points in the set of samples that are beyond any threshold, in any direction, concentrates appropriately to the expected fraction under the true distribution. The exact form of the concentration that we enforce here is necessary to get the sample complexities we desire. Observe that the bound on the difference gets stronger as T grows. Intuitively, this is possible because the expected value number of points from a Gaussian beyond this threshold gets smaller as T grows, which decreases the variance of the random variable. This allows us to apply correspondingly stronger Chernoff bounds, which give tighter concentration.

We show in Appendix E that a sufficiently large set of independent samples from $\mathcal{N}(\mu, I)$ satisfies these properties with high probability. Specifically, we prove:

Lemma 5.3.2. *Let $\varepsilon, \delta > 0$. Let S, n be as above, and let*

$$n = \Omega\left(\frac{d}{\varepsilon^2}\operatorname{poly}\log\left(\frac{d}{\varepsilon\delta}\right)\right) .$$

Then, S satisfies (5.7)–(5.10) *with probability $1 - \delta$.*

5.3.1 Filtering for robust isotropic mean estimation

Our main algorithmic contribution in this section is the design of a filtering algorithm, and a proof of its correctness under the deterministic conditions described above.

Our algorithm for this problem is *almost* of the form described in SpectralFilter, however does not quite fit for a (relatively dumb) technical reason. Recall that to define a filter, we need to define three conditions: a score function, a threshold function, and a removal function. We will describe these each in turn. Throughout this description, let $U \subseteq S$ be a (potentially already partially filtered) set of points, which we wish to filter.

Scores Morally, the scores we use are exactly those computed by ComputeSpectralScores, but we need to use a different algorithm for reasons we describe here.

Recall that ComputeSpectralScores finds an approximate top unit eigenvector v' of the empirical covariance $M^U(\mu^U)$, and defines the scores to be the squared correlation of each centered data point with this eigenvector. Instead here, we need to find a approximate top unit eigenvector v of $M^U(\mu^U) - I$, and we need to define the scores to be

$$\tau(X) = (v^\top (X - \mu^U))^2 - 1 \ ,$$

for every $X \in U$. If we do so, then we have

$$\begin{aligned}\mathop{\mathbb{E}}_{X \in_u U}[\tau(X)] &= \frac{1}{|U|} \sum_{X \in U} \left((v^\top (X - \mu^U))^2 - 1 \right) \\ &= v^\top M^U(\mu^U) v - 1 = v^\top \left(M^U(\mu^U) - I \right) v \ .\end{aligned}$$

Rather than in ComputeSpectralScores , here we need to do the spectral operations with respect to the empirical covariance minus the identity. This is because in this case we are going to exploit the fact that we know that the true covariance is the identity to detect spectral deviations to the empirical covariance of a relatively scale. Specifically, the deviations we will detect will be of the order $O(\varepsilon \log 1/\varepsilon)$.

This causes some difficulties if we try to directly plug in ComputeSpectralScores , because of the approximate nature of ApproxSVD. Instead, we will do approximate SVD directly on $M^U(\mu^U) - I$. The (approximate) pseudocode is given in Algorithm 16.

Algorithm 16 Computing scores via a spectral criterion for learning the mean of an isotropic sub-Gaussian distribution

```
1: function ComputeIsoScores($U$)
2:     Let $v$ be an (approximate) eigenvector of $M^U(\mu^U) - I$, i.e. $v$ satisfies
           $v^\top \left( M^U(\mu^U) - I \right) v \geq \frac{9}{10} \left\| M^U(\mu^U) - I \right\|_2$ .
3:     For $\tau \in U$, let $\tau(X) = (v^\top (X - \mu^U))^2 - 1$.
4:     return the function $\tau : U \to \mathbb{R}$
```

We remark that implementing Line 2 as written would require forming $M^U(\mu^U)$, which would be quite slow; much slower than the claimed runtime. However, the approximate top eigenvector of this matrix can still be computed using a minor modification to ApproxSVD. This is because each iteration of ApproxSVD only requires that are able to evaluate matrix-vector multiplications in nearly linear time. That is, for any vector $u \in \mathbb{R}^d$, we need to evaluate Mu in linear time, where M is the matrix we are applying ApproxSVD to. Since our matrix is $M^U(\mu^U) - I$, we can clearly do this without forming $M^U(\mu^U)$ explicitly. As a result, we have:

Corollary 5.3.3. *Given a dataset $U \subseteq \mathbb{R}^d$ of size m,* ComputeIsoScores(U) *runs in time $\widetilde{O}(md)$.*

For the rest of the section, we will for simplicity assume that the eigenvector found by ComputeIsoScores is the exact top eigenvector. It can be easily verified that none of our arguments change when we are only given an approximate top eigenvector.

Threshold In accordance with the general guidelines discussed, we may compute (5.5), with $f(x) = x^2 - 1$. In this case, reusing the notation of (5.5), we have that $\phi_v(x)$ is simply the PDF of an isotropic sub-Gaussian distribution, and $\Phi_v(1-\varepsilon) = O(\sqrt{\log 1/\varepsilon})$. Moreover, observe that by our choice of f, we have $E_{X \sim D_v}[f(X)] = 0$. Hence, we have

$$\begin{aligned}\mathfrak{T}_\varepsilon(D, f) &= \varepsilon \cdot \int_{O(\sqrt{\log 1/\varepsilon})}^{\infty} (x^2 - 1)\phi(x) dx \\ &\leq \varepsilon \int_{O(\sqrt{\log 1/\varepsilon})}^{\infty} x^2 \phi(x) dx \\ &= O\left(\varepsilon \log 1/\varepsilon\right) \ ,\end{aligned}$$

by standard Gaussian concentration. Thus, our threshold is simply to stop if the sum of the scores is greater than $O(\varepsilon \log 1/\varepsilon)$. The formal pseudocode is given in

Algorithm 17 Threshold function for learning the mean of an isotropic sub-Gaussian distribution

```
function IsoThres($\tau, \varepsilon, \delta$)
    return $\mathbb{E}_{X \in_u U}[\tau(X)] \leq O(\varepsilon \log 1/\varepsilon)$.
```

Removal The removal operation for this case is a bit subtle. Essentially, the idea will be to find a point $T > 0$ so that sub-Gaussian concentration is violated at this point; however, the specific form needs to be carefully worked out so as to work with the concentration inequality we have for LTFs, namely (5.10). We give the algorithm in Algorithm 18.

Algorithm 18 Removal function for learning the mean of an isotropic sub-Gaussian distribution

```
function IsoRemove($U, \tau, \varepsilon, \delta$)
    Let $C_1 = C_3 = 8$, and $C_2 = 1/2$. ▷ Constants for the tail bound which work in theory, but should be optimized in practice.
    Let $\rho := 3\sqrt{\varepsilon \, \mathbb{E}_{X \in_u U}[\tau(X)]}$. Find $T > 0$ such that
        $$\Pr_{X \in_u U}\left[|\tau(X)|^{1/2} > T + \rho\right] > C_1 \exp(-C_2 T^2) + C_3 \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\delta})\right)} .$$
    return the set $U' = \{X \in U : |\tau(X)|^{1/2} \leq T + \rho\}$.
```

The filter for isotropic sub-Gaussian distributions With these definitions, we now have a full algorithm for our filter algorithm. We will denote it

$$\begin{aligned}&\textsc{FilterIsoMean}(\cdot,\cdot,\cdot)\\ &\qquad := \textsc{GeneralFilter}(\cdot,\cdot,\cdot,\textsc{ComputeIsoScores},\textsc{IsoThres},\textsc{IsoRemove}) .\end{aligned}$$

Our main result about this algorithm is the following:

Proposition 5.3.4. *Let* (S, S') *satisfy* (5.6)–(5.10). *Let* $U \subseteq S'$ *be any set with* $\Delta(S, U) \leq \varepsilon$, *so that and for any* $X, Y \in U$, $\|X - Y\|_2 \leq O(\sqrt{d \log(d/\varepsilon\tau)})$. *Then given* U, ε, δ, FilterIsoMean *returns one of the following:*

(i) If FilterIsoMean *outputs "DONE", then* μ^U *satisfies* $\|\mu^U - \mu\|_2 = O(\varepsilon\sqrt{\log(1/\varepsilon)})$.

(ii) A set $U' \subseteq U$ *such that* $\Delta(S, U') \leq \Delta(S, U) - \frac{\varepsilon}{\alpha}$, *where*

$$\alpha = d \log\left(\frac{d}{\varepsilon\tau}\right) \log\left(d \log \frac{d}{\varepsilon\tau}\right) . \tag{5.11}$$

Moreover, the algorithm runs in time $\widetilde{O}(nd)$.

We pause briefly to interpret this proposition. The guarantee is that filtering will either: (1) output a good estimate of the true mean, or (2) decrease the fraction of bad points to good points in our current data set. We will show later in this section that, after first pruning the data set using NaivePrune (Fact 2.2.6), these conditions guarantee that if we iteratively apply FilterIsoMean to our data set, it will output a good estimate of the mean in at most $O(\alpha)$ iterations, where α is given in (5.11). Combined with the runtime guarantee for a single iteration of filtering, this guarantees that the algorithm will always output a good estimate of the true mean, in time at most $\widetilde{O}(nd^2)$, as required.

5.3.2 Proof of Proposition 5.3.4

In this section we prove Proposition 5.3.4. Observe that without loss of generality we may take $U = S'$, as the only condition we use about U is that $\Delta(S, U) \leq \varepsilon$.

In a slight abuse of notation, let $S_{\text{good}} = S \cap S'$ and $S_{\text{bad}} = S' \setminus S$, i.e., let them denote the sets of samples themselves, rather than the set of indices which are uncorrupted or corrupted, respectively. Moreover, let S_{rem} be the set of samples in S which have been removed in S'. Therefore $S' = (S \cup S_{\text{bad}}) \setminus S_{\text{rem}}$.

With this notation, we can write

$$\Delta(S, S') = \frac{|S_{\text{rem}}| + |S_{\text{bad}}|}{|S|} .$$

Thus, our assumption $\Delta(S, S') \leq \varepsilon$ is equivalent to $|S_{\text{rem}}| + |S_{\text{bad}}| \leq \varepsilon \cdot |S|$, and the definition of S' directly implies that $(1-\varepsilon)|S| \leq |S'| \leq (1+\varepsilon)|S|$. Throughout the proof, we assume that ε is a sufficiently small constant.

Throughout this section, for any $U \subseteq S \cup S'$ we wil let

$$M^U = M^U(\mu) = \mathop{\mathbb{E}}_{X \in_u U} \left[(X - \mu)(X - \mu)^\top \right] ,$$

that is, in (5.2) we will take $y = \mu$. Moreover, let

$$\widehat{\Sigma} = M^{S'}(\mu^{S'})$$

be the empirical covariance of the dataset.

Our analysis will hinge on proving the important claim that $\widehat{\Sigma}-I$ is approximately $(|S_{\text{bad}}|/|S'|)M^{S_{\text{bad}}}$. This means two things for us. First, it means that if the positive errors align in some direction (causing $M^{S_{\text{bad}}}$ to have a large eigenvalue), there will be a large eigenvalue in $\widehat{\Sigma}-I$. Second, it says that any large eigenvalue of $\widehat{\Sigma}-I$ will correspond to an eigenvalue of $M^{S_{\text{bad}}}$, which will give an explicit direction in which many error points are far from the empirical mean.

Formally, the key lemma we will prove is the following:

Lemma 5.3.5. *Let* (S,S') *satisfy* (5.6)–(5.10)*. Then*

$$\widehat{\Sigma}-I = \frac{|S_{\text{bad}}|}{|S'|}M^{S_{\text{bad}}} + O\left(\varepsilon \log(1/\varepsilon)\right) + O\left(\frac{|S_{\text{bad}}|}{|S'|}\right)^2 \|M^{S_{\text{bad}}}\|_2 \ ,$$

where the additive terms denote matrices of appropriately bounded spectral norm.

Proof of Lemma 5.3.5

We begin by noting that we have concentration bounds on Gaussians and therefore, on S.

Fact 5.3.6. *Let* (S,S') *satisfies* (5.6)-(5.10)*. Let* $w \in \mathbb{R}^d$ *be any unit vector, then for any* $T>0$*,*

$$\Pr_{X\sim\mathcal{N}(\mu,I)}\left[|w\cdot(X-\mu)|>T\right] \leq 2\exp(-T^2/2)$$

and

$$\Pr_{X\in_u S}\left[|w\cdot(X-\mu)|>T\right] \leq 2\exp(-T^2/2) + \frac{\varepsilon}{T^2\log\left(d\log(\frac{d}{\varepsilon\tau})\right)} \ .$$

Proof. The first line is Fact 1.4.1, and the former follows from (5.10). □

By using the above fact, we obtain the following simple claim:

Claim 5.3.7. *Let* (S,S') *satisfies* (5.6)-(5.10)*. Let* $w\in\mathbb{R}^d$ *be any unit vector, then* *for any $T > 0$, we have that:*

$$\Pr_{X \sim \mathcal{N}(\mu, I)}[|w \cdot (X - \mu^{S'})| > T + \|\mu^{S'} - \mu\|_2] \leq 2\exp(-T^2/2).$$

and

$$\Pr_{X \in_u S}[|w \cdot (X - \mu^{S'})| > T + \|\mu^{S'} - \mu\|_2] \leq 2\exp(-T^2/2) + \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\tau})\right)}.$$

Proof. This follows from Fact 5.3.6 upon noting that $|w \cdot (X - \mu^{S'})| > T + \|\mu^{S'} - \mu\|_2$ only if $|w \cdot (X - \mu)| > T$. □

We can use the above facts to prove concentration bounds for L. In particular, we have the following lemma:

Lemma 5.3.8. *Let (S, S') satisfies* (5.6)-(5.10)*. Then, we have that*

$$\|M^{S_{\text{rem}}}\|_2 = O\left(\log \frac{|S|}{|S_{\text{rem}}|} + \varepsilon \frac{|S|}{|S_{\text{rem}}|}\right) .$$

Proof. Since $S_{\text{rem}} \subseteq S$, for any $x \in \mathbb{R}^d$, we have that

$$|S| \cdot \Pr_{X \in_u S}(X = x) \geq |S_{\text{rem}}| \cdot \Pr_{X \in_u S_{\text{rem}}}(X = x) . \tag{5.12}$$

Since $M^{S_{\text{rem}}}$ is a symmetric matrix, we have $\|M^{S_{\text{rem}}}\|_2 = \max_{\|v\|_2 = 1} |v^\top M^{S_{\text{rem}}} v|$. So, to bound $\|M^{S_{\text{rem}}}\|_2$ it suffices to bound $|v^\top M^{S_{\text{rem}}} v|$ for unit vectors v. By definition of $M^{S_{\text{rem}}}$, for any $v \in \mathbb{R}^d$ we have that

$$|v^\top M^{S_{\text{rem}}} v| = \mathop{\mathbb{E}}_{X \in_u S_{\text{rem}}}[|v \cdot (X - \mu)|^2].$$

For unit vectors v, the RHS is bounded from above as follows:

$$\begin{aligned}
\mathop{\mathbb{E}}_{X \in_u S_{\text{rem}}}\left[|v \cdot (X-\mu)|^2\right] &= 2\int_0^{\infty} \Pr_{X \in_u S_{\text{rem}}}\left[|v \cdot (X-\mu)| > T\right] T dT \\
&= 2\int_0^{O(\sqrt{d\log(d/\varepsilon\tau)})} \Pr_{X \in_u S_{\text{rem}}}[|v \cdot (X-\mu)| > T] T dT \\
&\leq 2\int_0^{O(\sqrt{d\log(d/\varepsilon\tau)})} \min\left\{1, \frac{|S|}{|S_{\text{rem}}|} \cdot \Pr_{X \in_u S}\left[|v \cdot (X-\mu)| > T\right]\right\} T dT \\
&\leq \int_0^{4\sqrt{\log(|S|/|S_{\text{rem}}|)}} T dT \\
&+ (|S|/|S_{\text{rem}}|)\int_{4\sqrt{\log(|S|/|S_{\text{rem}}|)}}^{O(\sqrt{d\log(d/\varepsilon\tau)})} \Big(\exp(-T^2/2) + \frac{\varepsilon}{T^2 \log\left(d\log(\frac{d}{\varepsilon\tau})\right)}\Big) T dT \\
&= O\left(\log\frac{|S|}{|S_{\text{rem}}|} + \varepsilon \cdot \frac{|S|}{|S_{\text{rem}}|}\right) ,
\end{aligned}$$

where the second line follows from the fact that $\|v\|_2 = 1$, $S_{\text{rem}} \subset S$, and S satisfies (5.7); the third line follows from (5.12); and the fourth line follows from Fact 5.3.6.

□

As a corollary, we can relate the matrices $M^{S'}$ and $M^{S_{\text{bad}}}$, in spectral norm:

Corollary 5.3.9. *Let* (S, S') *satisfies* (5.6)-(5.10)*. Then, we have that*

$$M^{S'} - I = \frac{|S_{\text{bad}}|}{|S'|} M^{S_{\text{bad}}} + O(\varepsilon\log(1/\varepsilon)) ,$$

where the $O(\varepsilon\log(1/\varepsilon))$ *term denotes a matrix of spectral norm* $O(\varepsilon\log(1/\varepsilon))$*.*

Proof. By definition, we have that $|S'|M^{S'} = |S|M^S - |S_{\text{rem}}|M^{S_{\text{rem}}} + |S_{\text{bad}}|M^{S_{\text{bad}}}$. Thus, we can write

$$\begin{aligned}
M^{S'} &= (|S|/|S'|)M^S - (|S_{\text{rem}}|/|S'|)M^{S_{\text{rem}}} + (|S_{\text{bad}}|/|S'|)M^{S_{\text{bad}}} \\
&= I + O(\varepsilon) + O(\varepsilon\log(1/\varepsilon)) + (|S_{\text{bad}}|/|S'|)M^{S_{\text{bad}}} ,
\end{aligned}$$

where the second line uses the fact that $1 - 2\varepsilon \leq |S|/|S'| \leq 1 + 2\varepsilon$, (5.9), and Lemma 5.3.8. Specifically, Lemma 5.3.8 implies that $(|S_{\text{rem}}|/|S'|)\|M^{S_{\text{rem}}}\|_2 = O(\varepsilon\log(1/\varepsilon))$.

Therefore, we have that

$$M^{S'} - I = \frac{|S_{\mathrm{bad}}|}{|S'|} M^{S_{\mathrm{bad}}} + O(\varepsilon \log(1/\varepsilon)) \ ,$$

as desired. □

We now establish a similarly useful bound on the difference between the mean vectors:

Lemma 5.3.10. *Let (S, S') satisfies (5.6)-(5.10). We have that*

$$\mu^{S'} - \mu = \frac{|S_{\mathrm{bad}}|}{|S'|}(\mu^{S_{\mathrm{bad}}} - \mu) + O(\varepsilon\sqrt{\log(1/\varepsilon)}) \ ,$$

where the $O(\varepsilon\sqrt{\log(1/\varepsilon)})$ term denotes a vector with ℓ_2-norm at most $O(\varepsilon\sqrt{\log(1/\varepsilon)})$.

Proof. By definition, we have that

$$|S'|(\mu^{S'} - \mu) = |S|(\mu^S - \mu) - |S_{\mathrm{rem}}|(\mu^{S_{\mathrm{rem}}} - \mu) + |S_{\mathrm{bad}}|(\mu^{S_{\mathrm{bad}}} - \mu).$$

By (5.8) we have $\|\mu^S - \mu\|_2 = O(\varepsilon)$. Since $1 - 2\varepsilon \le |S|/|S'| \le 1 + 2\varepsilon$, it follows that

$$\frac{|S|}{|S'|}\|\mu^S - \mu\|_2 = O(\varepsilon) \ .$$

Lemma 2.2.16 implies $\|M^{S_{\mathrm{rem}}}\|_2 \ge \|\mu^{S_{\mathrm{rem}}} - \mu\|_2^2$. Together with Lemma 5.3.8, we obtain that

$$\|\mu^{S_{\mathrm{rem}}} - \mu\|_2 \le O\left(\sqrt{\log \frac{|S|}{|S_{\mathrm{rem}}|}} + \sqrt{\varepsilon \frac{|S|}{|S_{\mathrm{rem}}|}}\right) \ .$$

Therefore,

$$\frac{|S_{\mathrm{rem}}|}{|S'|}\|\mu^{S_{\mathrm{rem}}} - \mu\|_2 \le O\left(\frac{|S_{\mathrm{rem}}|}{|S|}\sqrt{\log \frac{|S|}{|L|}} + \sqrt{\varepsilon \frac{|L|}{|S|}}\right) = O(\varepsilon\sqrt{\log(1/\varepsilon)}) \ .$$

In summary,

$$\mu^{S'} - \mu = \frac{|S_{\mathrm{bad}}|}{|S'|}(\mu^{S_{\mathrm{bad}}} - \mu) + O(\varepsilon\sqrt{\log(1/\varepsilon)}) \ ,$$

as desired. This completes the proof of the lemma. □

We now the the tools necessary to prove Lemma 5.3.5:

Proof of Lemma 5.3.5. By definition, we can write

$$\widehat{\Sigma} - I = M^{S'} - I - (\mu^{S'} - \mu)(\mu^{S'} - \mu)^\top .$$

Using Corollary 5.3.9 and Lemma 5.3.10, we obtain:

$$\begin{aligned}
\widehat{\Sigma} - I &= \frac{|S_{\text{bad}}|}{|S'|} M^{S_{\text{bad}}} + O(\varepsilon \log(1/\varepsilon)) + O\left(\frac{|S_{\text{bad}}|}{|S'|}\right)^2 \|\mu^{S_{\text{bad}}} - \mu\|_2^2 + O(\varepsilon^2 \log(1/\varepsilon)) \\
&= \frac{|S_{\text{bad}}|}{|S'|} M^{S_{\text{bad}}} + O(\varepsilon \log(1/\varepsilon)) + O\left(\frac{|S_{\text{bad}}|}{|S'|}\right)^2 \|M^{S_{\text{bad}}}\|_2 ,
\end{aligned}$$

where the second line since Lemma 2.2.16 implies $\|M^{S_{\text{bad}}}\|_2 \geq \|\mu^E - \mu\|_2^2$. This completes the proof. □

Proof of Proposition 5.3.4 given Lemma 5.3.5

We now show how Lemma 5.3.5 implies Proposition 5.3.4. This entails demonstrating two things. We need to show that: (1) if the spectral norm of $\widehat{\Sigma} - I$ is small, i.e. when IsoThres returns True, then algorithm outputs a good mean, and (2) if the spectral norm of $\widehat{\Sigma} - I$ is large, then the algorithm throws out more bad points than good points in IsoRemove. We do these in turn.

Case of Small Spectral Norm. Suppose IsoThres outputs True. In this case, by the guarantees of ApproxSVD, we have that

$$\lambda^* := \|\widehat{\Sigma} - I\|_2 = O(\varepsilon \log(1/\varepsilon)) .$$

Hence, Lemma 5.3.5 yields that

$$\frac{|S_{\text{bad}}|}{|S'|} \|M^{S_{\text{bad}}}\|_2 \leq \lambda^* + O(\varepsilon \log(1/\varepsilon)) + O\left(\frac{|S_{\text{bad}}|}{|S'|}\right)^2 \|M^{S_{\text{bad}}}\|_2 ,$$

which in turns implies that

$$\frac{|S_{\mathrm{bad}}|}{|S'|}\|M^{S_{\mathrm{bad}}}\|_2 = O(\varepsilon \log(1/\varepsilon)) \ .$$

On the other hand, since $\|M^{S_{\mathrm{bad}}}\|_2 \geq \|\mu^{S_{\mathrm{bad}}} - \mu\|_2^2$, Lemma 5.3.10 gives that

$$\|\mu^{S'} - \mu\|_2 \leq \frac{|S_{\mathrm{bad}}|}{|S'|}\sqrt{\|M^{S_{\mathrm{bad}}}\|_2} + O(\varepsilon\sqrt{\log(1/\varepsilon)}) = O(\varepsilon\sqrt{\log(1/\varepsilon)}).$$

This proves part (i) of Proposition 5.3.4.

Case of Large Spectral Norm. We next show the correctness of the algorithm when it returns a filter in Step 3.

We start by proving that if $\lambda^* := \|\widehat{\Sigma} - I\|_2 > C\varepsilon \log(1/\varepsilon)$, for a sufficiently large universal constant C, then a value T satisfying the condition in Step 3 exists. We first note that $\|M^{S_{\mathrm{bad}}}\|_2$ is appropriately large. Indeed, by Lemma 5.3.5, the guarantees of ComputeIsoScores, and the assumption that $\lambda^* > C\varepsilon \log(1/\varepsilon)$ we deduce that

$$\frac{|S_{\mathrm{bad}}|}{|S'|}\|M^{S_{\mathrm{bad}}}\|_2 = \Omega(\lambda^*) \ . \tag{5.13}$$

Moreover, using the inequality $\|M^{S_{\mathrm{bad}}}\|_2 \geq \|\mu^{E} - \mu\|_2^2$ and Lemma 5.3.10 as above, we get that

$$\|\mu^{S'} - \mu\|_2 \leq \frac{|S_{\mathrm{bad}}|}{|S'|}\sqrt{\|M^{S_{\mathrm{bad}}}\|_2} + O(\varepsilon\sqrt{\log(1/\varepsilon)}) \leq \delta/2 \ , \tag{5.14}$$

where we used the fact that $\delta =: \sqrt{\varepsilon \lambda^*} > C'\varepsilon\sqrt{\log(1/\varepsilon)}$.

Let v^* denote the top eigenvector of $\widehat{\Sigma} - I$, so that $|\tau(X)|^{1/2} = \left|v^* \cdot (X - \mu^{S'}\right|$. Suppose for the sake of contradiction that for all $T > 0$ we have that

$$\Pr_{X \in_u S'}\left[|\tau(X)|^{1/2} > T + \delta\right] \leq 8\exp(-T^2/2) + 8\frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\tau})\right)} \ .$$

Using (5.14), this implies that for all $T > 0$ we have that

$$\Pr_{X \in_u S'} [|v^* \cdot (X - \mu)| > T + \delta/2] \leq 8 \exp(-T^2/2) + 8 \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\tau})\right)} . \tag{5.15}$$

Since $S_{\text{bad}} \subseteq S'$, for all $x \in \mathbb{R}^d$ we have that

$$|S'| \Pr_{X \in_u S'}[X = x] \geq |S_{\text{bad}}| \Pr_{Y \in_u S_{\text{bad}}}[Y = x] .$$

This fact combined with (5.15) implies that for all $T > 0$

$$\Pr_{X \in_u S_{\text{bad}}} [|v^* \cdot (X - \mu)| > T + \delta/2] \leq C \frac{|S'|}{|S_{\text{bad}}|} \left(\exp(-T^2/2) + \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\tau})\right)} \right) , \tag{5.16}$$

for some universal constant C''.

We now have the following sequence of inequalities:

$$\begin{aligned}
\|M^{S_{\text{bad}}}\|_2 &= \mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}} \left[|v^* \cdot (X - \mu)|^2 \right] = 2 \int_0^\infty \Pr_{X \in_u S_{\text{bad}}} [|v^* \cdot (X - \mu)| > T] \, T dT \\
&= 2 \int_0^{O(\sqrt{d \log(d/\varepsilon\tau)})} \Pr_{X \in_u S_{\text{bad}}} [|v^* \cdot (X - \mu)| > T] \, T dT \\
&\leq 2 \int_0^{O(\sqrt{d \log(d/\varepsilon\tau)})} \min \left\{ 1, \frac{|S'|}{|S_{\text{bad}}|} \cdot \Pr_{X \in_u S'} \left[|v^* \cdot (X - \mu)| > T \right] \right\} T dT \\
&\leq \int_0^{4\sqrt{\log(|S'|/|S_{\text{bad}}|)} + \delta} T dT \\
&\quad + C'' \frac{|S'|}{|S_{\text{bad}}|} \int_{4\sqrt{\log(|S'|/|S_{\text{bad}}|)} + \delta}^{O(\sqrt{d \log(d/\varepsilon\tau)})} \left(\exp(-T^2/2) + \frac{\varepsilon}{T^2 \log\left(d \log(\frac{d}{\varepsilon\tau})\right)} \right) T dT \\
&\leq \int_0^{4\sqrt{\log(|S'|/|S_{\text{bad}}|)} + \delta} T dT \\
&\quad + C'' \frac{|S'|}{|S_{\text{bad}}|} \left(\int_{4\sqrt{\log(|S'|/|S_{\text{bad}}|)} + \delta}^{\infty} \left(\exp(-T^2/2) \right) T dT + O(\varepsilon) \right) \\
&\leq \log \frac{|S'|}{|S_{\text{bad}}|} + \delta^2 + O(1) + O(\varepsilon) \cdot \frac{|S'|}{|S_{\text{bad}}|} \\
&\leq \log \frac{|S'|}{|S_{\text{bad}}|} + \varepsilon\lambda^* + O(\varepsilon) \cdot \frac{|S'|}{|S_{\text{bad}}|} \; .
\end{aligned}$$

Rearranging the above, we get that

$$\frac{|S_{\text{bad}}|}{|S'|}\|M^{S_{\text{bad}}}\|_2 \leq \frac{|S_{\text{bad}}|}{|S'|}\log\frac{|S'|}{|S_{\text{bad}}|} + \frac{|S_{\text{bad}}|}{|S'|}\varepsilon\lambda^* + O(\varepsilon) = O(\varepsilon\log(1/\varepsilon) + \varepsilon^2\lambda^*).$$

Combined with (5.13), we obtain $\lambda^* = O(\varepsilon\log(1/\varepsilon))$, which is a contradiction if C is sufficiently large. Therefore, it must be the case that for some value of T the condition in Step 3 is satisfied.

The following claim completes the proof:

Claim 5.3.11. *Fix $\alpha =: d\log(d/\varepsilon\tau)\log(d\log(\frac{d}{\varepsilon\tau}))$. We have that $\Delta(S,U') \leq \Delta(S,S') - 2\varepsilon/\alpha$.*

Proof. Recall that $S' = (S \setminus S_{\text{rem}}) \cup S_{\text{bad}}$, with S_{bad} and S_{rem} disjoint sets such that $S_{\text{rem}} \subset S$. We can similarly write $U' = (S \setminus {S_{\text{rem}}}') \cup {S_{\text{bad}}}'$, with ${S_{\text{rem}}}' \supseteq S_{\text{rem}}$ and ${S_{\text{bad}}}' \subseteq S_{\text{bad}}$. Since

$$\Delta(S,S') - \Delta(S,U') = \frac{|S_{\text{bad}} \setminus {S_{\text{bad}}}'| - |{S_{\text{rem}}}' \setminus S_{\text{rem}}|}{|S|},$$

it suffices to show that

$$|S_{\text{bad}} \setminus {S_{\text{bad}}}'| \geq |{S_{\text{rem}}}' \setminus S_{\text{rem}}| + \varepsilon\frac{|S|}{\alpha} .$$

Note that $|{S_{\text{rem}}}' \setminus S_{\text{rem}}|$ is the number of points rejected by the filter that lie in $S \cap S'$. Note that the fraction of elements of S that are removed to produce S'' (i.e., satisfy $|v^* \cdot (x - \mu^{S'})| > T + \delta$) is at most $2\exp(-T^2/2) + \varepsilon/\alpha$. This follows from Claim 5.3.7 and the fact that $T = O(\sqrt{d\log(d/\varepsilon\tau)})$.

Hence, it holds that $|{S_{\text{rem}}}' \setminus S_{\text{rem}}| \leq (2\exp(-T^2/2) + \varepsilon/\alpha)|S|$. On the other hand, Step 3 of the algorithm ensures that the fraction of elements of S' that are rejected by the filter is at least $8\exp(-T^2/2) + 8\varepsilon/\alpha)$. Note that $|S_{\text{bad}} \setminus {S_{\text{bad}}}'|$ is the number of points rejected by the filter that lie in $S' \setminus S$. Therefore, we can write:

$$\begin{aligned} |S_{\text{bad}} \setminus {S_{\text{bad}}}'| &\geq (8\exp(-T^2/2) + 8\varepsilon/\alpha)|S'| - (2\exp(-T^2/2) + \varepsilon/\alpha)|S| \\ &\geq (8\exp(-T^2/2) + 8\varepsilon/\alpha)|S|/2 - (2\exp(-T^2/2) + \varepsilon/\alpha)|S| \\ &\geq (2\exp(-T^2/2) + 3\varepsilon/\alpha)|S| \\ &\geq |{S_{\text{rem}}}' \setminus S_{\text{rem}}| + 2\frac{\varepsilon|S|}{\alpha} \ , \end{aligned}$$

where the second line uses the fact that $|S'| \geq |S|/2$ and the last line uses the fact that $|{S_{\text{rem}}}' \setminus S_{\text{rem}}|/|S| \leq 2\exp(-T^2/2) + \varepsilon/\alpha$. Noting that $\log(d/\varepsilon\tau) \geq 1$, this completes the proof of the claim. □

5.3.3 Putting it all together

We finish by showing how Theorem 5.3.1 follows easily from Proposition 5.3.4. Given Algorithm FilterIsoMean, our algorithm is simple: we first run NaivePrune, then run FilterIsoMean until it outputs an estimate of the mean.

Algorithm 19 Filtering algorithm for agnostically learning the mean.

```
function LearnMeanFilter($\varepsilon, \delta, X_1, \ldots, X_n$)
    Run NaivePrune($X_1, \ldots, X_n$). Let $S' = \{X_i\}_{i \in I}$ be the pruned set of samples.
    while true do
        Run FilterIsoMean($S', \varepsilon, \delta$).
        if FilterIsoMean($S', \varepsilon, \delta$) outputs "DONE" then
            break
        else
            Let $S' \leftarrow$ FilterIsoMean($S', \varepsilon, \delta$)
    return $\mu^{S'}$
```

Proof of Theorem 5.3.1. By the definition of $\Delta(S, S')$, since S' has been obtained from S by corrupting an ε-fraction of the points in S, we have that $\Delta(S, S') \leq 2\varepsilon$. By Lemma 5.3.2, we have that (S, S') satisfy conditions (5.6)-(5.10) with probability $1 - \delta$. We henceforth condition on this event.

By (5.7), we have that have $\|X - \mu\|_2 \leq O(\sqrt{d \log |S|/\tau})$ for all $X \in S$. Thus, the NaivePrune procedure does not remove from S' any member of S. Hence, its output, S'', has $\Delta(S, S'') \leq \Delta(S, S')$ and for any $X \in S''$, there is a $X \in S$ with $\|X - Y\|_2 \leq O(\sqrt{d \log |S|/\tau})$. By the triangle inequality, for any $X, Z \in S''$, $\|X - Z\|_2 \leq O(\sqrt{d \log |S|/\tau}) = O(\sqrt{d \log(d/\varepsilon\tau)})$.

Then, we iteratively apply the FilterIsoMean procedure of Proposition 5.3.4 until it terminates, in which case LearnMeanFilter outputs a mean vector $\widehat{\mu}$ with $\|\widehat{\mu} - \mu\|_2 = O(\varepsilon\sqrt{\log(1/\varepsilon)})$. We claim that we need at most $O(\alpha)$ iterations for this to happen. Indeed, the sequence of iterations results in a sequence of sets S'_i, such that $\Delta(S, S'_i) \leq \Delta(S, S') - i \cdot \varepsilon/\alpha$. Thus, if we do not output the empirical mean in the first 2α iterations, in the next iteration there are no outliers left. Hence in the next iteration it is impossible for the algorithm to output a subset satisfying Condition (ii) of Proposition 5.3.4, so it must output a mean vector satisfying (i), as desired. □

5.4 Learning a Gaussian With unknown covariance

In this section, we use our filter technique to agnostically learn a Gaussian with zero mean vector and unknown covariance. By combining the algorithms of the current and the previous subsections, as in our convex programming approach (Section 2.2.4), we obtain a filter-based algorithm to agnostically learn an arbitrary unknown Gaussian.

The main result of this subsection is the following theorem:

Theorem 5.4.1. *Let $\varepsilon, \delta > 0$, and let $\Sigma \succ 0$. Let S' be an ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$ of size n, where*

$$n = \Omega\left(\frac{d^2}{\varepsilon^2} \operatorname{poly}\log(d/\varepsilon\delta)\right) .$$

There exists an efficient algorithm that, given S', ε and δ, returns $\widehat{\Sigma}$ so that with probability at least $1 - \delta$, it holds $\|\Sigma - \widehat{\Sigma}\|_\Sigma = O(\varepsilon \log(1/\varepsilon))$. Moreover, the algorithm runs in time $\widetilde{O}(\varepsilon n^2 d^2)$.

5.4.1 Additional preliminaries

The following definition will also be convenient for us:

Definition 5.4.1. For any $d \geq 1$, and any $d \times d$ matrix $\Sigma \succ 0$, we let $\mathcal{P}_2(\Sigma)$ denote the set of even degree-2 polynomials $p : \mathbb{R}^d \to \mathbb{R}$ so that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,\Sigma)}[p(X)] = 0 \text{ and } \mathop{\mathrm{Var}}_{X \sim \mathcal{N}(0,\Sigma)}[p(X)] = 1 .$$

In a slight abuse of notation, throughout this section, given a dataset U, we will let

$$M^U = M^U(0) = \mathop{\mathbb{E}}_{X \in_u U}\left[XX^\top\right] ,$$

that is, we will assume by default that $y = 0$ in (5.2). This is for convenience since in this section we will always assume the mean is zero.

Deterministic conditions

Throughout this section, we will let $\Sigma \succ 0$ denote the (unknown) true covariance matrix. As in the previous section, we will need a condition on S under which our algorithm will succeed. As in Definition 5.1.1, $\Delta(S, S')$ is the size of the symmetric difference of S and S' divided by $|S|$.

Specifically, we fix $\varepsilon, \delta > 0$, and let S, S' be subsets of points in $\mathbb{R}^d$. We will assume that (S, S') satisfy:

$$\Delta(S, S') \leq \varepsilon \tag{5.17}$$

$$X^\top \Sigma^{-1} X < O(d \log(|S|/\delta)) \text{ ,} \forall X \in S \tag{5.18}$$

$$\left\| \mathop{\mathbb{E}}_{X \in_u S}\left[XX^\top\right] - \Sigma \right\|_\Sigma = O(\varepsilon) , \tag{5.19}$$

$$\mathop{\mathrm{Var}}_{X \in_u S}[p(X)] = (1 \pm O(\varepsilon)) \cdot \mathop{\mathrm{Var}}_{X \sim \mathcal{N}(0,I)}[p(X)] . \tag{5.20}$$

$$\Pr_{X \in_u S}\left[|p(X)| > T\right] \leq \frac{\varepsilon}{T^2 \log^2(T)} \text{ for all } p \in \mathcal{P}_2(\Sigma) \text{ and } T > 10 \log 1/\varepsilon . \tag{5.21}$$

Let us first note some basic properties of such polynomials on a normal distribution. The proof of this lemma is deferred to Section E.

Lemma 5.4.2. *For any even degree-2 polynomial $p : \mathbb{R}^d \to \mathbb{R}$, we can write $p(x) = (\Sigma^{-1/2}x)^\top P_2(\Sigma^{-1/2}x) + p_0$, for a $d \times d$ symmetric matrix P_2 and $p_0 \in \mathbb{R}$. Then, for $X \sim \mathcal{N}(0, \Sigma)$, we have*

1. $\mathbb{E}[p(X)] = p_0 + \mathrm{tr}(P_2)$,

2. $\mathrm{Var}[p(X)] = 2\|P_2\|_F^2$ *and*

3. *For all $T > 1$,* $\Pr(|p(X) - \mathbb{E}[p(X)]| \geq T) \leq 2e^{1/3-2T/3\,\mathrm{Var}[p(X)]}$.

4. *For all $\rho > 0$,* $\Pr(|p(X)| \leq \rho^2) \leq O(\rho)$.

We note that, if S is obtained by taking random samples from $\mathcal{N}(0, \Sigma)$, then S satisfies (5.18)-(5.21) with high probability. The proof of this lemma is also deferred to Section E.

Lemma 5.4.3. *Let $\varepsilon, \delta > 0$, and let S be a set of n samples from $\mathcal{N}(0, \Sigma)$, where*

$$n = \Omega\left(\frac{d^2 \log^5(d/(\varepsilon\delta))}{\varepsilon^2}\right) .$$

Then S satisfies (5.18)-(5.21) *with probability* $1 - \delta$.

The basic thrust of our algorithm is as follows: By Lemma 5.4.3, with high probability we have that S is (ε, δ)-good with respect to G. The algorithm is then handed a new set S' such that $\Delta(S, S') \leq 2\varepsilon|S|$. The algorithm will run in stages. In each stage, the algorithm will either output good estimates for the covariance, or will return a new set S'' such that $\Delta(S, S'') < \Delta(S, S')$. In the latter case, the algorithm will recurse on S''. As before, the key algorithmic component to this algorithm, will be the design of the filtering algorithm which we repeatedly run.

5.4.2 Filtering for robust covariance estimation

In this section we design a filtering algorithm for covariance estimation, and a proof of its correctness given the determinstic conditions given above.

Our goal will be to either obtain a certificate that the empirical covariance of our current data set is close to the true covariance, or to devise a filter that allows us to clean up our data set by removing some elements, most of which are corrupted.

The idea here is the following. Let (S, S') satisfy (5.17)-(5.21), and let $M' = M^{S'}$ be the empirical covariance of S'. We know by Corollary 1.4.6 that $\mathcal{N}(0, \Sigma)$ and $\mathcal{N}(0, M')$ are close unless $I - \Sigma^{-1/2} M' \Sigma^{-1/2}$ has large Frobenius norm. This happens if and only if there is some matrix A with $\|A\|_F = 1$ such that

$$\mathrm{tr}(A\Sigma^{-1/2} M' \Sigma^{-1/2} - A) = \mathop{\mathbb{E}}_{X \in_u U}[(\Sigma^{-1/2}X)^\top A(\Sigma^{-1/2}X) - \mathrm{tr}(A)]$$

is far from 0. On the other hand, we know that the distribution of

$$p(X) = (\Sigma^{-1/2}X)^\top A(\Sigma^{-1/2}X) - \mathrm{tr}(A)$$

for $X \in_u S$ is approximately that of $p(X)$ when $X \sim \mathcal{N}(0, \Sigma)$. In order to substantially change the mean of this function, while only changing S at a few points, one must have several points in S' for which $p(X)$ is abnormally large. This in turn will imply that the variance of $p(X)$ for X from S' must be large. This phenomenon will be detectable as a large eigenvalue of the matrix of fourth moments of $X \in S'$ (thought of as a matrix over the space of second moments). If such a large eigenvalue is detected, we will have a p with $p(X)$ having large variance. By throwing away from S' elements for which $|p(X)|$ is too large after some appropriate centering, we will return a cleaner version of S'.

Scores Motivated by this definition, our scores should be given by this polynomial p which has large variance over the dataset. This can be found via spectral methods on the fourth moment tensor. The formal method is given in ComputeCovScores.

For technical reasons (we will need to appropriately center the τ later on), we will return the non-squared scores, even though the direct analogy with the previous algorithms would suggest the alternative score function $\tau = p^*(X)^2$.

The following lemma parses the guarantees of ComputeCovScores, and states

Algorithm 20 Algorithm for finding a filtering polynomial

```
function ComputeCovScores($U$)
    Let $M = M^U$ and let $n = |U|$
    Compute an eigen-decomposition of $M$ and use it to compute $M^{-1/2}$
    Let $X_1, \ldots, X_n$ be the elements of $U$.
    For $i = 1, \ldots, n$, let $Y_i = M^{-1/2} X_{(i)}$ and $Z_i = Y_i^{\otimes 2}$.
    Let
        $$T = \frac{1}{|S'|} \sum_{i=1}^{n} Z_i Z_i^\top - \left(I^\flat\right)\left(I^\flat\right)^\top .$$
    Approximate the top eigenvalue $\lambda^*$ and corresponding eigenvector $v^*$ of $T$ restricted to $\mathcal{S}_{\mathrm{sym}}$
    Let $p^*(x) = \frac{1}{\sqrt{2}}((M^{-1/2}x)^\top v^{*\sharp}(M^{-1/2}x) - \mathrm{tr}(v^{*\sharp}))$.
    return the function $\tau(X) = p^*(X)$
```

that it indeed finds the polynomial we wanted:

Claim 5.4.4. *Let U be any set of points, and let $M = M^U$. Let $p^* =$ ComputeCovScores(U). Then we have*

$$p^* = \operatorname*{arg\,max}_{p \in \mathcal{P}_2(M)} \mathop{\mathbb{E}}_{X \in_u U} \left[p(X)^2\right] .$$

Proof. Let $p \in \mathcal{P}_2(M)$ be arbitrary. By Lemma 5.4.2 all even polynomials with degree-2 that have $\mathbb{E}_{X \sim \mathcal{N}(0,M)}[p(X)] = 0$ can be written as $p(x) = (M^{-1/2}x)^\top P_2(M'^{-1/2}x) - \mathrm{tr}(P_2)$ for a symmetric matrix P_2. If we take $P_2 = v^\sharp/\sqrt{2}$ for a unit vector v such that $v^\sharp$ is symmetric, then $\mathrm{Var}_{X \sim \mathcal{N}(0,M)}[p(X)] = 2\|P_2\|_F = \|v_2\| = 1$. Hence any polynomial output by ComputeCovScores will be in $\mathcal{P}_2(M)$, as claimed.

We now show that the output of ComputeCovScores is the maximizer of the quadratic form claimed. Note that since the second moment matrix of U is M, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{X\in_u U}[p(X)] &= \mathop{\mathbb{E}}_{X\in_u U}\left[(M^{-1/2}X)^\top P_2(M^{-1/2}X) - \mathrm{tr}(P_2)\right] \\ &= \mathop{\mathbb{E}}_{X\in_u U}\left[\mathrm{tr}((XX^\top)M^{-1/2}P_2M^{-1/2})\right] - \mathrm{tr}(P_2) \\ &= \mathrm{tr}\left(\mathop{\mathbb{E}}_{X\in_u U}[(XX^\top)]M^{-1/2}P_2M^{-1/2}\right) - \mathrm{tr}(P_2) \\ &= \mathrm{tr}(MM^{-1/2}P_2M^{-1/2}) - \mathrm{tr}(P_2) = 0\ .\end{aligned}$$

We let $V = \{\Sigma^{-1/2}X : X \in U\}$, and we let $W = \{Y^{\otimes 2} : Y \in T\}$. We thus have

$$\begin{aligned}\mathop{\mathbb{E}}_{X\in_u U}[p(X)^2] &= \mathop{\mathbb{E}}_{Y\in_u V}[(Y^\top P_2 Y - \mathrm{tr}(P_2))^2] \\ &= \mathop{\mathbb{E}}_{Y\in_u V}[(Y^\top P_2 Y)^2] + \mathrm{tr}(P_2)^2 - 2\mathrm{tr}(P_2))^2] \\ &= \mathop{\mathbb{E}}_{Y\in_u V}[\mathrm{tr}(((YY^\top)P_2)^2] - \mathrm{tr}(P_2 I)^2 - 0 \\ &= \frac{1}{2}\mathop{\mathbb{E}}_{Z\in_u W}[(Z^\top v)^2] - \frac{1}{2}(v^\top I^\flat)^2 \\ &= \frac{1}{2}\left(\mathop{\mathbb{E}}_{Z\in_u W}[v^\top(ZZ^\top)v] - 2v^\top(I^\flat I^{\flat T})v\right) \\ &= \frac{1}{2}v^\top T v\ .\end{aligned}$$

Thus, the $p(x)$ that maximizes $\mathbb{E}_{X\in_u U}[p(X)^2]$ is given by the unit vector v that maximizes $v^\top T v$ subject to $v^\sharp$ being symmetric. Since ComputeCovScores exactly finds the top eigenvector of T subject to this constraint, this demonstrates that if p^* is the output of ComputeCovScores, then we have

$$p^* = \mathop{\arg\max}_{p\in\mathcal{P}_2(M)} \mathop{\mathbb{E}}_{X\in_u U}\left[p(X)^2\right]\ ,$$

as claimed. □

The function ComputeCovScores uses similar notation to SeparationOracleUnknownCovariance, so that they can be more easily compared. Indeed, observe that the ultimate form of the score function (being the top eigenvalue of a fourth moment tensor) is essentially the same as the form of the separating hyperplane in FilterGaussianCov. The major difference is that in FilterGaussianCov for technical reasons, we restrict ourselves first to a subspace (to remove the $(I^\flat)(I^\flat)^\top)$ term, whereas here we do not.

Threshold Since this doesn't quite fit into the framework of spectral filtering, we cannot use the exact calculation as done in Section 5.2.2. However, we may apply the same principles. Notice that by the arguments above, when choosing the thresholds, what we really care about is not $\tau(X)$, but rather $\tau(X)^2$, as this gives us the top eigenvector of the fourth moment matrix.

For any $p \in \mathcal{N}(0, \Sigma)$, by definition the expected value of $p^2(X)$ under $\mathcal{N}(0, \Sigma)$ is 1. Thus the question is, how much can the largest ε-fraction of values of $p(X)$ contribute in aggregate? But by Gaussian concentration (specifically Hanson-Wright), it is not hard to show that this value is $O(\varepsilon \log^2 1/\varepsilon)$. Therefore our threshold should be $1 + O(\varepsilon \log^2 1/\varepsilon)$. This exactly gives the threshold we use in CovThres:

Algorithm 21 Threshold function for learning the covariance of a mean zero Gaussian.

```
function CovThres($\tau, \varepsilon, \delta$)
    return $\mathbb{E}_{X \in_u U}[\tau(X)^2] \leq 1 + O(\varepsilon \log^2 1/\varepsilon)$.
```

Removal As before, the specific form of the tail bound we will use is a bit subtle here. This is again necessary so that we can work with the types of concentration guarantees that we have available for degree-2 PTFs.

The filter for robust covariance estimation We now have the tools to describe the full filtering algorithm. Recall that for robust mean estimation, the algorithm worked in two parts: first it did a naive pruning step, then ran the iterative filtering algorithm until completion. However, here it turns out the make more sense (at least in theory) to simultaneously prune and filter. This is because as we get a better estimate of the covariance, more and more points may become "obvious" outliers. The formal pseudocode for the algorithm is given in Algorithm 23

Algorithm 22 Removal function for learning the covariance of a mean-zero distribution

```
function CovRemove($U, \tau, \varepsilon, \delta$)
    Let $C$ be a sufficiently large constant.
    Let $\mu$ be the median value of $\tau(X)$ over $X \in U$.
    Find a $T \geq C$ such that
        $$\Pr_{X \in_u U}(|\tau(X) - \mu| \geq T + 3) \geq \mathrm{Tail}(T, \varepsilon) ,$$
    where
        $$\mathrm{Tail}(T, \varepsilon) = \begin{cases} 3\varepsilon/(T^2 \log^2(T)) & \text{if } T \geq 10 \ln(1/\varepsilon); \\ 1 & \text{otherwise.} \end{cases}$$
    return $U' = \{X \in U : |\tau(X) - \mu| < T\}$.
```

Algorithm 23 Filter algorithm for a Gaussian with unknown covariance matrix.

```
procedure FilterGaussianCov($U, \varepsilon, \delta$)
    Let $C > 0$ be sufficiently large universal constants.
    Let $M' \leftarrow \mathbb{E}_{X \in_u U}[X X^\top]$.
    if there is any $X \in U$ such that $X^\top (M')^{-1} X \geq C d \log(|S'|/\delta)$ then
        return $U' = U \setminus \{X \in U : X^\top (M')^{-1} X \geq C d \log(|S'|/\delta)\}$.
    else
        return the output of
    GeneralFilter($U, \varepsilon, \delta$, ComputeCovScores, CovThres, CovRemove) .
```

Our main correctness claim is the following:

Proposition 5.4.5. *Let $\varepsilon, \delta > 0$ be fixed, and let (S, S') satisfy* (5.17)-(5.21), *where $n = |S'|$. Let $U \subseteq S'$ with $\Delta(S, U) < \varepsilon$. Then given U, ε, δ,* FilterGaussianCov *returns one of the following:*

(i) If FilterGaussianCov *outputs "DONE", then Σ^U satisfies $\|\Sigma^U - \Sigma\|_F = O(\varepsilon \log(1/\varepsilon))$.*

(ii) A set $U' \subseteq U$ such that $\Delta(S, U') < \Delta(S, U)$.

Moreover, the algorithm runs in time $\widetilde{O}(nd^2 + d^\omega)$.

The remainder of this section is dedicated to a proof of Proposition 5.4.5.

Runtime of FilterGaussianCov

We first make some remarks about the runtime of the algorithm. Forming Σ' can be done in time $O(nd^2)$ by naive methods, and inverting it can be done in d^ω time. The remaining operations except for ComputeCovScores only involve evaluating a quadratic polynomial on the samples and simple sorting operations, and so can also be done in time $O(nd^2)$. Thus it remains to implement ComputeCovScores in $\widetilde{O}(nd^2)$ time.

In ComputeCovScores, forming the Z_i can be done in time $O(nd^2)$. As for robust mean estimation, it is easily verified that it suffices to find a constant approximation to the top eigenvector of M, i.e., it would suffice to simply find any vector v so that

$$v^\top T v \geq (1 - \varepsilon) \max_{\|v\|_2 = 1} v^\top T v \; .$$

Thus, naively we would like to apply ApproxPCA to find the approximate top eigenvalue and eigenvector of M. But M does not easily factor into the form which ApproxPCA immediately applies. But as we discussed for ComputeIsoScores, we observe that ApproxPCA simply requires us to efficiently evaluate matrix-vector products. Since given the $Z_1, \ldots, Z_n$, matrix-vector products with M can be done in time $O(nd^2)$, we can implement ApproxPCA in time $\widetilde{O}(nd^2)$. Thus, overall the algorithm runs in time $\widetilde{O}(nd^2 + d^\omega)$, as claimed.

Analysis of Algorithm 23: Proof of Proposition 5.4.5

We now show the correctness of FilterGaussianCov. In the subsequent sections, we will assume that ComputeCovScores finds an exact maximizer of the flattened fourth moment tensor. As mentioned previously, it is easy to verify that the following arguments trivially extend to the case when we have an approximate maximizer. We will also always assume that all calls to ApproxPCA succeed. As before, by paying an additional polylogarithmic overhead, this occurs except with negligible probability. This provides an alternative interpretation of the top eigenvalue of the fourth moment tensor that is so critical to the methods in the previous chapters: it shows that the top eigenvector of these tensors that we find corresponds exactly to the quadratic polynomial that maximizes a standardized quadratic form, which corresponds to some direction where the variance of some polynomial will be too large under the empirical distribution.

With this in hand, we now begin to argue correctness. As before, it suffices to argue the case when $U = S'$, as the argument extends straightforwardly to general U satisfying the conditions in the theorem. As for robust mean estimation, we write $S' = (S \setminus S_{\mathrm{rem}}) \cup S_{\mathrm{bad}}$, and we let $S_{\mathrm{good}} = S \setminus S_{\mathrm{rem}}$. It is then the case that

$$\Delta(S, S') = \frac{|S_{\mathrm{rem}}| + |S_{\mathrm{bad}}|}{|S|} \; .$$

Since this is small we have that $|S_{\mathrm{rem}}|, |S_{\mathrm{bad}}| = O(\varepsilon |S'|)$. For conciseness we also let $M' = M^{S'}$. Observe that

$$M' = \frac{|S_{\mathrm{good}}|}{|S'|} M^{S_{\mathrm{good}}} + \frac{|S_{\mathrm{bad}}|}{|S'|} M^{S_{\mathrm{bad}}} = M^{S_{\mathrm{good}}} + O(\varepsilon)(M^{S_{\mathrm{bad}}} - M^{S_{\mathrm{good}}}) \; .$$

A critical part of our analysis will be to note that $M^{S_{\mathrm{good}}}$ is very close to Σ, and thus that either Σ' is very close to Σ or else $M^{S_{\mathrm{bad}}}$ is very large in some direction.

Lemma 5.4.6. *Let S satisfy* (5.18)-(5.21). *We have that*

$$\|I - \Sigma^{-1/2} M^{S_{\mathrm{good}}} \Sigma^{-1/2}\|_F = O(\varepsilon \log(1/\varepsilon).$$

To prove Lemma 5.4.6, we will require the following:

Lemma 5.4.7. *Let S satisfy* (5.18)-(5.21). *Let $p \in \mathcal{P}_2(\Sigma)$. Then, we have that*

$$|S_{\mathrm{rem}}| \cdot \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{rem}}} [p(X)^2] = O(\varepsilon \log^2(1/\varepsilon) |S|) \; , \; \textit{and} \tag{5.22}$$

$$|S_{\mathrm{rem}}| \cdot \left| \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{rem}}} [p(X)] \right| = O(\varepsilon \log(1/\varepsilon) |S|) \; . \tag{5.23}$$

Proof. This holds essentially because the distribution of $p(X)$ for $X \in S$ is close to that for $p(X)$ for $X \sim \mathcal{N}(0, \Sigma)$, which has rapidly decaying tails. Therefore, throwing away an ε-fraction of the mass cannot change the value of the variance by very much. In particular, we have that

$$\begin{aligned}
|S_{\mathrm{rem}}| \cdot \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{rem}}}[p(X)^2] &= \int_0^\infty |S_{\mathrm{rem}}| \Pr_{X \in_u S_{\mathrm{rem}}}(|p(X)| > T) 2T dT \\
&\leq \int_0^\infty |S| \min\left(\varepsilon, \Pr_{X \in_u S}(|p(X)| > T)\right) 2T dT \\
&\leq \int_0^{10 \ln(1/\varepsilon)} 4\varepsilon |S| T dT + \int_{10 \ln(1/\varepsilon)}^\infty 6|S|\varepsilon T/(T^2 \log^2(T)) dT \\
&\leq O(\varepsilon |S| \log^2(1/\varepsilon)) + \int_{10 \ln(1/\varepsilon)}^\infty 6|S|\varepsilon/(T \log^2(T)) dT \\
&= O(\varepsilon |S| \log^2(1/\varepsilon)) + 6\varepsilon |S| / \ln(10 \ln(1/\varepsilon)) \\
&= O(\varepsilon \log^2(1/\varepsilon) |S|) \; .
\end{aligned}$$

This proves (5.22). To prove (5.23), observe that by the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\frac{|S_{\mathrm{rem}}|}{|S|} \cdot \left| \mathop{\mathbb{E}}_{x \in_u S_{\mathrm{rem}}}[p(X)] \right| &\leq \frac{|S_{\mathrm{rem}}|}{|S|} \sqrt{\mathop{\mathbb{E}}_{x \in_u S_{\mathrm{rem}}}[p(X)^2]} \\
&\leq \sqrt{\frac{|S_{\mathrm{rem}}|}{|S|}} \cdot \sqrt{O(\varepsilon \log^2(1/\varepsilon))} = O(\varepsilon \log(1/\varepsilon) \; ,
\end{aligned}$$

as desired. □

Now we can prove Lemma 5.4.6.

Proof of Lemma 5.4.6. Note that, since the matrix inner product is an inner product,

$$\| I - \Sigma^{-1/2} M^{S_{\mathrm{good}}} \Sigma^{-1/2} \|_F = \sup_{\|A\|_F = 1} \left(\mathrm{tr}(A \Sigma^{-1/2} M^{S_{\mathrm{good}}} \Sigma^{-1/2}) - \mathrm{tr}(A) \right) .$$

We need to show that for any A with $\|A\|_F = 1$ that $\mathrm{tr}(A \Sigma^{-1/2} M^{S_{\mathrm{good}}} \Sigma^{-1/2}) - \mathrm{tr}(A)$ is small.

Since

$$\begin{aligned}\mathrm{tr}(A\Sigma^{-1/2}M^{S_{\mathrm{good}}}\Sigma^{-1/2}) &= \mathrm{tr}(A^\top\Sigma^{-1/2}M^{S_{\mathrm{good}}}\Sigma^{-1/2}) \\ &= \mathrm{tr}\left(\frac{1}{2}(A+A^\top)\Sigma^{-1/2}M^{S_{\mathrm{good}}}\Sigma^{-1/2}\right) ,\end{aligned}$$

and $\|\frac{1}{2}(A+A^\top)\|_F \leq \frac{1}{2}(\|A\|_F + \|A^\top\|_F) = 1$, we may assume WLOG that A is symmetric.

Consider such an A. We note that

$$\begin{aligned}\mathrm{tr}(A\Sigma^{-1/2}M^{S_{\mathrm{good}}}\Sigma^{-1/2}) &= \mathop{\mathbb{E}}_{X\in_u S_{\mathrm{good}}}[\mathrm{tr}(M\Sigma^{-1/2}XX^\top\Sigma^{-1/2})] \\ &= \mathop{\mathbb{E}}_{X\in_u S_{\mathrm{good}}}[(\Sigma^{-1/2}X)^\top A(\Sigma^{-1/2}X)] .\end{aligned}$$

Let $p(x)$ denote the quadratic polynomial

$$p(x) = (\Sigma^{-1/2}x)^\top A(\Sigma^{-1/2}x) - \mathrm{tr}(A) .$$

By Lemma 5.4.2, $\mathbb{E}_{X\sim\mathcal{N}(0,\Sigma)}[p(X)] = 0$ and $\mathrm{Var}_{X\sim\mathcal{N}(0,\Sigma)}[p(X)] = 2\|A\|_F^2 = 2$.

By (5.19), we have that

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{X\in_u S}[p(X)]\right| &= \left|\mathop{\mathbb{E}}_{X\in_u S}[p(X)] - \mathop{\mathbb{E}}_{X\sim\mathcal{N}(0,\Sigma)}[p(X)]\right| \\ &= \left|\mathrm{tr}\left(A\Sigma^{-1/2}\left(\mathop{\mathbb{E}}_{X\in_u S}[XX^\top]-\Sigma\right)\Sigma^{-1/2}\right)\right| \\ &\overset{(a)}{\leq} 2\left\|\Sigma^{-1/2}\left(\mathop{\mathbb{E}}_{X\in_u S}[XX^\top]-\Sigma\right)\Sigma^{-1/2}\right\|_F \\ &= 2\left\|\mathop{\mathbb{E}}_{X\in_u S}[XX^\top]-\Sigma\right\|_\Sigma \leq O(\varepsilon) ,\end{aligned}$$

where (a) follows from self-duality of the Frobenius norm, and since $\|A\|_F = 2$. Therefore, it suffices to show that the contribution from L is small. In particular, it will be enough to show that

$$\frac{|S_{\mathrm{rem}}|}{|S|}|\mathop{\mathbb{E}}_{x\in_u S_{\mathrm{rem}}}[p(X)]| \leq O(\varepsilon\log(1/\varepsilon)).$$

This follows from Lemma 5.4.7, which completes the proof. □

As a corollary of this we note that Σ' cannot be too much smaller than Σ.

Corollary 5.4.8. *Let* (S, S') *satisfy* (5.17)-(5.21). *Then, we have*

$$M' \succeq (1 - O(\varepsilon \log(1/\varepsilon)))\Sigma \ .$$

Proof. Lemma 5.4.6 implies that $\Sigma^{-1/2} M^{S_{\text{good}}} \Sigma^{1/2}$ has all eigenvalues in the range $1 \pm O(\varepsilon \log(1/\varepsilon)$. Therefore, $M^{S_{\text{good}}} \succeq (1 + O(\varepsilon \log(1/\varepsilon)))\Sigma$. Our result now follows from noting that

$$M' = \frac{|S_{\text{good}}|}{|S|} M^{S_{\text{good}}} + \frac{|S_{\text{bad}}|}{|S|} M^{S_{\text{bad}}} \ ,$$

and $M^{S_{\text{bad}}} \succeq 0$. □

The first step in verifying correctness is to note that if our algorithm returns on Step 5 that it does so correctly.

Claim 5.4.9. *Let* (S, S') *satisfy* (5.17)-(5.21). *Then, if* FilterGaussianCov *returns on Step 5, then* $\Delta(S, U') < \Delta(S, U)$.

Proof. This is clearly true if we can show that all X removed have $X \notin S$. However, this follows because Corollary 5.4.8 implies that $(M')^{-1} \preceq 2\Sigma^{-1}$, and therefore, by (5.18), we have

$$X^\top (M')^{-1} X \leq 2 X^\top \Sigma^{-1} X < C d \log(N/\delta)$$

for all $X \in S$, and for C sufficiently large. □

Next, we need to show that if our algorithm returns "DONE", then we have $\|M' - \Sigma\|_\Sigma$ is small.

Claim 5.4.10. *Let* (S, S') *satisfy* (5.17)-(5.21). *If our algorithm returns "DONE", then* $\|M' - \Sigma\|_\Sigma = O(\varepsilon \log 1/\varepsilon)$.

Proof. We note that

$$\begin{aligned}\|I - \Sigma^{-1/2} M' \Sigma^{-1/2}\|_F &\leq \|I - \Sigma^{-1/2} M^{S_{\text{good}}} \Sigma^{-1/2}\|_F + \frac{|S_{\text{bad}}|}{|S'|} \|I - \Sigma^{-1/2} M^{S_{\text{bad}}} \Sigma^{-1/2}\|_F \\ &\leq O(\varepsilon \log(1/\varepsilon)) + \frac{|S_{\text{bad}}|}{|S'|} \|I - \Sigma^{-1/2} M^{S_{\text{bad}}} \Sigma^{-1/2}\|_F \ ,\end{aligned}$$

where the last line follows from Lemma 5.4.6. Therefore, we will have an appropriate bound unless $\|I - \Sigma^{-1/2} M^{S_{\text{bad}}} \Sigma^{-1/2}\|_F = \Omega(\log(1/\varepsilon))$.

Next, note that there is a symmetric matrix A with $\|A\|_F = 1$ such that

$$\begin{aligned}\|I - \Sigma^{-1/2} M^{S_{\text{bad}}} \Sigma^{-1/2}\|_F &= \operatorname{tr}(A \Sigma^{-1/2} M^{S_{\text{bad}}} \Sigma^{-1/2} - A) \\ &= \mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}} [(\Sigma^{-1/2} X)^\top A (\Sigma^{-1/2} X) - \operatorname{tr}(A)] \ .\end{aligned}$$

Let $p(X)$ be the polynomial

$$p(X) = \frac{1}{\sqrt{2}} \left((\Sigma^{-1/2} X)^\top A (\Sigma^{-1/2} X) - \operatorname{tr}(A) \right) \ ,$$

so that

$$\mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}} [p(X)] = \frac{1}{\sqrt{2}} \mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}} [(\Sigma^{-1/2} X)^\top A (\Sigma^{-1/2} X) - \operatorname{tr}(A)] \ .$$

Using Lemma 5.4.2, $\mathbb{E}_{X \sim \mathcal{N}(0,\Sigma)}[p(X)] = 0$ and $\operatorname{Var}_{X \sim \mathcal{N}(0,\Sigma)}[p(X)] = 1$. Therefore, $p \in \mathcal{P}_2(\Sigma)$. Therefore, since our algorithm returned at this step, by Claim 5.4.4, we have that $\mathbb{E}_{X \in_u U}[p(X)^2] \leq 1 + O(\varepsilon)$. Moreover, by Lemma 5.4.7, we have $|S_{\text{rem}}| \cdot \mathbb{E}_{X \in_u S_{\text{rem}}}[p(X)^2] \leq O(\varepsilon \log^2(1/\varepsilon))|S|$.

Therefore, we have

$$\begin{aligned}(1 + O(\varepsilon))|U| &= |U| \cdot \mathop{\mathbb{E}}_{X \in_u U}[p(X)^2] \\ &= |S| \cdot \mathop{\mathbb{E}}_{X \in_u S}[p(X)^2] - |S_{\text{rem}}| \cdot \mathop{\mathbb{E}}_{X \in_u S_{\text{rem}}}[p(X)^2] + |S_{\text{bad}}| \cdot \mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}}[p(X)^2] \\ &= (1 + O(\varepsilon)) \cdot |S| + O(\varepsilon \log^2(1/\varepsilon))|S| + |S_{\text{bad}}| \cdot \mathop{\mathbb{E}}_{X \in_u S_{\text{bad}}}[p(X)^2] \ ,\end{aligned}$$

where the last line follows from (5.19) and Lemma 5.4.7.

Simplifying, and using the fact that $|U|/|S| \geq (1-\varepsilon)$, this implies that

$$|S_{\mathrm{bad}}| \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{bad}}}[p(X)^2] = O(\varepsilon \log^2(1/\varepsilon))|S| \; .$$

Thus, by Cauchy-Schwarz, and since $|S_{\mathrm{bad}}|/|S| \leq \varepsilon$, we have

$$\left| \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{bad}}}[p(X)] \right| \leq \sqrt{\mathop{\mathbb{E}}_{X \in_u S_{\mathrm{bad}}}[p(X)^2]} \leq O(\log 1/\varepsilon) \; ,$$

as desired. This shows that if the algorithm returns in this step, it does so correctly.

□

Next, we need to show that if the algorithm reaches Step 4 that such a T exists.

Claim 5.4.11. *Let (S, S') satisfy (5.17)-(5.21). If the algorithm reaches Step 4, then there exists a $T > 1$ such that*

$$\Pr_{X \in_u S'}(|p(X) - \mu| \geq T) \geq 12 \exp(-(T-1)/3) + 3\varepsilon/(d \log(n/\delta))^2.$$

Proof. Before we begin, we will need the following critical Lemma:

Lemma 5.4.12. *Let (S, S') satisfy (5.17)-(5.21). If the algorithm reaches Step 4, then*

$$\mathop{\mathrm{Var}}_{X \sim \mathcal{N}(0,\Sigma)}[p^*(X)] \leq 1 + O(\varepsilon \log(1/\varepsilon)).$$

Proof. We note that since $\mathrm{Var}_{X \sim \mathcal{N}(0,M')}(p(X)) = 1$, we just need to show that the variance with respect to $\mathcal{N}(0, \Sigma)$ instead of $\mathcal{N}(0, M')$ is not too much larger. This will essentially be because Σ cannot be much bigger than the covariance matrix of M by Corollary 5.4.8.

Recall that p^* is the polynomial in $\mathcal{P}_2(M')$ which maximizes the variance of the empirical covariance, and $\tau(X) = p^*(X)$. Using Lemma 5.4.2, we can write

$$p^*(x) = (M'^{-1/2}x)^\top P_2(M'^{-1/2}x) + p_0 \; ,$$

where $\|P_2\|_F = \frac{1}{2}\operatorname{Var}_{X\sim\mathcal{N}(0,M')}(p(X)) = \frac{1}{2}$ and $p_0 = \mu + \operatorname{tr}(P_2)$. We can also express $p^*(x)$ in terms of Σ as $p^*(x) = (\Sigma^{-1/2}x)^\top M(\Sigma^{-1/2}x) + p_0$, and have $\operatorname{Var}_{X\sim G}[p(X)] = \|M\|_F$. Here, M is the matrix $\Sigma^{1/2}M'^{-1/2}P_2M'^{-1/2}\Sigma^{1/2}$. By Corollary 5.4.8, it holds $M' \geq (1 - O(\varepsilon\log(1/\varepsilon)))\Sigma$. Consequently, $\Sigma^{1/2}M'^{-1/2} \leq (1 + O(\varepsilon\log(1/\varepsilon)))I$, and so $\|\Sigma^{1/2}M'^{-1/2}\|_2 \leq 1 + O(\varepsilon\log(1/\varepsilon))$. Similarly, $\|M'^{-1/2}\Sigma^{1/2}\|_2 \leq 1 + O(\varepsilon\log(1/\varepsilon))$.

We claim that if A, B are matrices, then $\|AB\|_F \leq \|A\|_2\|B\|_F$. If B_j are the columns of B, then we have $\|AB\|_F^2 = \sum_j \|AB_j\|_2^2 \leq \|A\|_2^2 \sum_j \|B_j\|_2^2 = (\|A\|_2\|B\|_F)^2$. Similarly for rows, we have $\|AB\|_F \leq \|A\|_F\|B\|_2$.

Thus, we have

$$\begin{aligned}\operatorname*{Var}_{X\sim\mathcal{N}(0,\Sigma)}[p^*(X)] &= 2\|M\|_F \\ &\leq 2\|\Sigma^{1/2}M'^{-1/2}\|_2\|P_2\|_F\|M'^{-1/2}\Sigma^{1/2}\|_2 \\ &\leq 1 + O(\varepsilon\log(1/\varepsilon)) \ .\end{aligned}$$

□

Next, we need to consider the deviation due to the fact that we are using μ rather than the true mean of p. However we claim that this deviation cannot contribute too much to the difference. In particular, we note that by the similarity of S and S', μ must be between the $50-\varepsilon$ and $50+\varepsilon$ percentiles of values of $p^*(X)$ for $X \in S$. However, since S satisfies (5.21), this must be between the $50-2\varepsilon$ and $50+2\varepsilon$ percentiles of $p^*(X)$ for $X \sim \mathcal{N}(0,\Sigma)$. Therefore, by Cantelli' s inequality,

$$|\mu| \leq 2\sqrt{\operatorname*{Var}_{X\sim\mathcal{N}(0,\Sigma)}[p^*(X)]} \leq 3 \ , \tag{5.24}$$

since $\mathbb{E}_{X\sim\mathcal{N}(0,\Sigma)}[p^*(X)] = 0$. We are now ready to proceed. Our argument will follow by noting that while the variance of p^* is much larger than expected, very little of this discrepancy can be due to points in S_{good}. Therefore, the points of S_{bad} must provide a large contribution. Given that there are few points in S_{bad}, much of this contribution must come from there being many points near the tails, and this will guarantee that some valid threshold T exists.

In particular, we have that $\mathrm{Var}_{X \in_u S'}(p^*(X) \geq 1 + C\varepsilon \log^2(1/\varepsilon)$, which means that

$$\frac{\sum_{X \in S'} |p(X) - \hat{\mu}|^2}{|S'|} \geq \operatorname*{Var}_{X \in_u S'}[p^*(X)] \geq 1 + C\varepsilon \ln^2(1/\varepsilon) \; .$$

Now, by (5.20), we know that

$$\begin{aligned} \frac{\sum_{X \in S} |p^*(X) - \hat{\mu}|^2}{|S|} &= \operatorname*{\mathbb{E}}_{X \sim \mathcal{N}(0,\Sigma)}[|p^*(X) - \hat{\mu}|^2](1 + O(\varepsilon)) \\ &= \operatorname*{Var}_{X \sim \mathcal{N}(0,\Sigma)}[p^*(X)](1 + O(\varepsilon)) \\ &\leq 1 + O(\varepsilon \log(1/\varepsilon)) \; , \end{aligned}$$

where the last line follows since $p^* \in \mathcal{P}_2(\Sigma)$. Therefore, since $\Delta(S, S') \leq \varepsilon$, we have that

$$\frac{\sum_{X \in S_{\mathrm{good}}} |p^*(X) - \hat{\mu}|^2}{|S'|} \leq 1 + O(\varepsilon \log(1/\varepsilon)) \; ,$$

as well. Hence, for C sufficiently large, it must be the case that

$$\sum_{X \in S_{\mathrm{bad}}} |p^*(X) - \hat{\mu}|^2 \geq \frac{C}{2} \varepsilon \log^2(1/\varepsilon)|S'| = \Omega(\log^2(1/\varepsilon) \cdot |S_{\mathrm{bad}}|) \; ,$$

and therefore, by (5.24), we have

$$\sum_{X \in S_{\mathrm{bad}}} |p^*(X) - \mu|^2 \geq \frac{C}{3} \varepsilon \log^2(1/\varepsilon) \cdot |S'| \; .$$

On the other hand, we have that

$$\begin{aligned} \sum_{X \in S_{\mathrm{bad}}} |p^*(X) - \mu|^2 &= \int_0^\infty \{X \in S_{\mathrm{bad}} : |p^*(X) - \mu| > T\} 2T dT \\ &\leq \int_0^{C^{1/4}\log(1/\varepsilon)} O(T\varepsilon|S'|)dt + \int_{C^{1/4}\ln(1/\varepsilon)}^\infty \{X \in S_{\mathrm{bad}} : |p^*(X) - \mu| > T\} 2T dT \\ &\leq O(C^{1/2}\varepsilon \log^2(1/\varepsilon)|S'|) + |S'| \int_{C^{1/4}\log(1/\varepsilon)}^\infty \Pr_{X \in_u S'}(|p^*(X) - \mu| > T) 2T dT \; . \end{aligned}$$

Therefore, we have that

$$\int_{C^{1/4}\log(1/\varepsilon)}^{\infty} \Pr_{X \in_u S'}(|p^*(X) - \mu| > T) 2T dT \geq \frac{C}{4}\varepsilon \log^2(1/\varepsilon) \ . \tag{5.25}$$

Assume for sake of contradiction that

$$\Pr_{X \in_u S'}(|p^*(X) - \mu| \geq T + 3) \leq \mathrm{Tail}(T, \varepsilon) \ ,$$

for all $T > 1$. Then, we would have that

$$\begin{aligned} \int_{10\log(1/\varepsilon)+3}^{\infty} \Pr_{X \in_u S'}(|p^*(X) - \mu| > T) 2T dT &\leq \int_{10\log(1/\varepsilon)}^{\infty} \frac{6(T+3)\varepsilon}{T^2 \log^2 T} dT \\ &= \int_{10\log(1/\varepsilon)}^{\infty} \frac{8\varepsilon}{T \log^2 T} dT \\ &= \frac{8\varepsilon}{\log(10\log(1/\varepsilon))} \ . \end{aligned}$$

For a sufficiently large C, this contradicts Equation (5.25). □

Finally, we need to verify that if our algorithm returns output in Step 5, that it is correct.

Claim 5.4.13. *If the algorithm returns during Step 5, then $\Delta(S, U') < \Delta(S, S')$.*

Proof. We note that it is sufficient to show that $|S_{\mathrm{bad}} \backslash U'| > |S_{\mathrm{good}} \backslash U'|$. In particular, it suffices to show that

$$|\{X \in S_{\mathrm{bad}} : |p^*(X) - \mu| > T + 3\}| > |\{X \in S_{\mathrm{good}} : |p^*(X) - \mu| > T + 3\}| \ .$$

For this, it suffices to show that

$$|\{X \in S' : |p^*(X) - \mu| > T + 3\}| > 2|\{X \in S_{\mathrm{good}} : |p^*(X) - \mu| > T + 3\}| \ ,$$

or that

$$|\{X \in S' : |p^*(X) - \mu| > T + 3\}| > 2|\{X \in S : |p^*(X) - \mu| > T + 3\}| \ .$$

By assumption, we have that

$$|\{X \in S' : |p^*(X) - \mu| > T + 3\}| > \frac{3\varepsilon |S'|}{T^2 \log^2 T} \; .$$

On the other hand, using(5.24) and (5.21), we have

$$\begin{aligned} |\{X \in S : |p^*(X) - \mu| > T + 3\}| &\leq |\{X \in S : |p^*(X) - \hat{\mu}| > T\}| \\ &\leq \frac{\varepsilon |S| \varepsilon}{T^2 \log^2 T} \; . \end{aligned}$$

This completes our proof. □

5.4.3 Putting it all together: proof of Theorem 5.4.1

Given Proposition 5.4.5, the full algorithm and proof of correctness are quite easy. The algorithm simply repeatedly applies FilterGaussianCov until it outputs "DONE", at which point we simply output the empirical second moment of the remaining data set. The formal algorithm description is given in Algorithm 24. We now demonstrate that Algorithm 24 gives the desired guarantees.

Algorithm 24 Filtering algorithm for agnostically learning the covariance.

```
function LearnCovarianceFilter($\varepsilon, \delta, X_1, \ldots, X_n$)
    while true do
        Run FilterGaussianCov($S', \varepsilon, \delta$).
        if it outputs "DONE" then
            break
        else
            Let $S' \leftarrow$ FilterGaussianCov($S', \varepsilon, \delta$)
    return $M^{(S')}$
```

Proof of Theorem 5.4.1. By Lemma 5.4.3 the original set S is (ε, δ)-good with respect to G with probability at least $1 - \delta$. Then, (S', S) satisfies the hypotheses of Proposition 5.4.5. We then repeatedly iterate the algorithm from Proposition 5.4.5 until it outputs a distribution G' close to G. This must eventually happen because at every step the distance between S and the set returned by the algorithm decreases by at least 1. Moreover, since the algorithm removes at least one corrupted data point each iteration, the algorithm cannot run for more than εn iterations. Combined with the per-iteration runtime guarantees of Proposition 5.4.5, this yields the claimed runtime. □

5.5 Learning the mean with bounded second moment

In this section, we use our filtering technique to give a near sample-optimal computationally efficient algorithm to robustly estimate the mean of a density with a second moment assumption. We show:

Theorem 5.5.1. *Let P be a distribution on $\mathbb{R}^d$ with unknown mean vector μ and unknown covariance matrix $\Sigma \preceq \sigma^2 I$. Let S be an ε-corrupted set of samples from P of size n, where*

$$n = \Omega\left(\frac{d \log d}{\varepsilon}\right) .$$

Then there exists an algorithm that given σ, ε, S, with probability $2/3$, outputs $\widehat{\mu}$ with $\|\widehat{\mu} - \mu\|_2 \leq O(\sigma\sqrt{\varepsilon})$ in time $\mathrm{poly}(d/\varepsilon)$.

Observe that without loss of generality we may assume $\sigma = 1$, as we can simply scale the points down by σ, then scale the result back, and obtain the desired result.

The algorithm for doing this will be the first usage of SpectralFilter. The most notable algorithmic difference between the algorithm for this instance and for the sub-Gaussian case is that the removal step will be randomized. Instead of looking for a deterministic violation of a concentration inequality, here we will choose a threshold *at random* (with a bias towards higher thresholds). The reason is that, in this setting, the spectral scores will be a constant fraction larger than what they should be. Therefore, randomly choosing a threshold weighted towards higher thresholds suffices to throw out more corrupted samples than uncorrupted samples *in expectation*. Although it is possible to reject many good samples this way, the algorithm still only rejects a total of $O(\varepsilon)$ samples with high probability. Interestingly, to the best of our knowledge, it seems that this randomness is necessary to get the right rates. Because of the weaker concentration of the data points that we have in this setting, deterministic conditions more akin to those used for the sub-Gaussian case seem to lose dimension-dependent factors. We leave it as an interesting open question if this is necessary or not.

Deterministic conditions As is tradition, we give a set of deterministic conditions under which our algorithm will work. Throughout this section, let P be the unknown distribution with unknown mean μ and unknown covariance $\Sigma \preceq I$. We would like our good set of samples to have mean close to that of P and bounded variance in all directions. However, we will have to be a bit careful: it turns out that since we have no assumptions about higher moments, it may be be possible for points from the true distribution to affect our sample covariance too much. Fortunately, such outliers have small probability and do not contribute too much to the mean, so we will later reclassify them as errors. This motivates the following definition:

Definition 5.5.1. We call a set S ε-good for a distribution P with mean μ and covariance $\Sigma \preceq I$ if the mean μ^S and covariance Σ^S of S satisfy $\|\mu^S - \mu\|_2 \leq \sqrt{\varepsilon}$ and $\|M^S(\mu^S)\|_2 \leq 2$.

We first show that given a set of i.i.d. points from P, there exists a large set of good points:

Lemma 5.5.2. *Let S be a set of*

$$n = \Theta\left(\frac{d \log d}{\varepsilon}\right)$$

samples drawn from P. Then, with probability at least $9/10$, a random $X \in_u S$ satisfies

(i) $\|\mathbb{E}_S[X] - \mu\|_2 \leq \sqrt{\varepsilon}/3$,

(ii) $\Pr_S\left[\|X - \mu\|_2 \geq 80\sqrt{d/\varepsilon}\right] \leq \varepsilon/160$,

(iii) $\left\|\mathbb{E}_S\left[(X-\mu)\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}\right]\right\|_2 \leq \sqrt{\varepsilon}/3$, *and*

(iv) $\left\|\mathbb{E}_S\left[(X-\mu)(X-\mu)^T\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}\right]\right\|_2 \leq 3/2.$

Proof. For (i), note that

$$\mathop{\mathbb{E}}_S[\|\mathbb{E}[X]-\mu\|_2^2] = \sum_i \mathop{\mathbb{E}}_S[(\mathbb{E}[X]_i-\mu_i)^2] \leq d/N \leq \varepsilon/360 \ ,$$

and so by Markov's inequality, with probability at least 39/40, we have $\|\mathbb{E}[X]-\mu\|_2^2 \leq \varepsilon/9$.

For (ii), similarly to (i), note that

$$\mathbb{E}[\|Y-\mu\|_2^2] = \sum_i \mathbb{E}\left[(Y_i-\mu_i)^2\right] \leq d \ ,$$

for $Y \sim P$. By Markov's inequality, $\Pr[\|Y-\mu\|_2 \geq 80\sqrt{d/\varepsilon}] \leq \varepsilon/160$ with probability at least 39/40.

For (iii), let $\nu = \mathbb{E}_{X\sim P}[X\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}]$ be the true mean of the distribution when we condition on the event that $\|X-\mu\|_2 \leq 80\sqrt{d/\varepsilon}$. By the same argument as (i), we know that

$$\left\|\mathop{\mathbb{E}}_{X\in_u S}\left[X\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}\right]-\nu\right\|_2 \leq \sqrt{\varepsilon}/9 \ ,$$

with probability at least 39/40. Thus it suffices to show that $\left\|\nu-\mu\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}\right\|_2 \leq \sqrt{\varepsilon}/10$. To do so, it suffices to show that for all unit vectors $v\in\mathbb{R}^d$, we have

$$\left|\left\langle v, \nu-\mu\cdot 1_{\|X-\mu\|_2\leq 80\sqrt{d/\varepsilon}}\right\rangle\right| < \sqrt{\varepsilon}/10 \ .$$

Observe that for any such v, we have

$$\begin{aligned}\left\langle v, \mu \cdot 1_{\|X-\mu\|_2 \leq 80\sqrt{d/\varepsilon}} - \nu \right\rangle &= \mathop{\mathbb{E}}_{X \sim P}\left[\langle v, X - \mu\rangle \cdot 1_{\|X-\mu\|_2 \leq 80\sqrt{d/\varepsilon}}\right] \\ &\overset{(a)}{\leq} \sqrt{\mathop{\mathbb{E}}_{X\sim P}[\langle v, X-\mu\rangle^2] \Pr_{X\sim P}[\|X-\mu\|_2 \geq 80\sqrt{d/\varepsilon}]} \\ &\overset{(b)}{=} \sqrt{v^T \Sigma v \cdot \Pr_{X\sim P}\left[\|X-\mu\|_2 \geq 80\sqrt{d/\varepsilon}\right]} \\ &\overset{(c)}{\leq} \sqrt{\varepsilon}/10 \ ,\end{aligned}$$

where (a) follows from Cauchy-Schwarz, and (b) follows from the definition of the covariance, and (c) follows from the assumption that $\Sigma \preceq I$ and from Markov's inequality.

For (iv), we require the following Matrix Chernoff bound:

Lemma 5.5.3 (Theorem 5.1.1 of [Tro15]). *Consider a sequence of $d \times d$ positive semi-definite random matrices X_k with $\|X_k\|_2 \leq L$ for all k. Let $\mu^{\max} = \|\sum_k \mathbb{E}[X_k]\|_2$. Then, for $\theta > 0$,*

$$\mathbb{E}\left[\left\|\sum_k X_k\right\|_2\right] \leq (e^\theta - 1)\mu^{\max}/\theta + L\log(d)/\theta \ ,$$

and for any $\delta > 0$,

$$\Pr\left[\left\|\sum_k X_k\right\|_2 \geq (1+\delta)\mu^{\max}\right] \leq d(e^\delta/(1+\delta)^{1+\delta})^{\mu^{\max}/L} \ .$$

We apply this lemma with $X_k = (x_k - \mu)(x_k - \mu)^T 1_{\|x_k - \mu\|_2 \leq 80\sqrt{d/\varepsilon}}$ for $\{x_1, \ldots, x_N\} = S$. Note that $\|X_k\|_2 \leq (80)^2 d/\varepsilon = L$ and that $\mu^{\max} \leq N\|\Sigma_P\|_2 \leq N$.

Suppose that $\mu^{\max} \leq N/80$. Then, taking $\theta = 1$, we have

$$\mathbb{E}[\left\|\sum_k X_k\right\|_2] \leq (e-1)N/80 + O(d\log(d)/\varepsilon) \ .$$

By Markov's inequality, except with probability 39/40, we have $\|\sum_k X_k\|_2 \leq N +$

$O(d\log(d)/\varepsilon) \leq 3N/2$, for N a sufficiently high multiple of $d\log(d)/\varepsilon$.

Suppose that $\mu^{\max} \geq N/80$, then we take $\delta = 1/2$ and obtain

$$\Pr\left[\left\|\sum_k X_k\right\|_2 \geq 3\mu^{\max}2\right] \leq d(e^{3/2}/(5/2)^{3/2})^{N\varepsilon/20d} .$$

For N a sufficiently high multiple of $d\log(d)/\varepsilon$, we get that $\Pr[\left\|\sum_k X_k\right\|_2 \geq 3\mu^{\max}/2] \leq 1/40$. Since $\mu^{\max} \leq N$, we have with probability at least $39/40$, $\left\|\sum_k X_k\right\|_2 \leq 3N/2$.

Noting that $\left\|\sum_k X_k\right\|_2/N = \|\mathbb{E}[1_{\|X-\mu\|_2 \leq 80\sqrt{d/\varepsilon}}(X-\mu)(X-\mu)^T]\|_2$, we obtain (iv). By a union bound, (i)-(iv) all hold simultaneously with probability at least $9/10$. □

Now we can get a 2ε-corrupted good set from an ε-corrupted set of samples satisfying Lemma 5.5.2, by reclassifying outliers as errors:

Lemma 5.5.4. *Let $S = R \cup E \setminus L$, where R is a set of $N = \Theta(d\log d/\varepsilon)$ samples drawn from P and E and L are disjoint sets with $|E|, |L| \leq \varepsilon$. Then, with probability* $9/10$, *we can also write $S = S_{\text{good}} \cup S_{\text{bad}} \setminus S_{\text{rem}}$, where $S_{\text{good}} \subseteq R$ is ε-good, $S_{\text{rem}} \subseteq L$ and $E \subseteq S_{\text{bad}}$ has $|S_{\text{bad}}| \leq 2\varepsilon|S|$.*

Proof. Let $S_{\text{good}} = \{x \in R : \|x\|_2 \leq 80\sqrt{d/\varepsilon}\}$. Condition on the event that R satisfies Lemma 5.5.2. By Lemma 5.5.2, this occurs with probability at least $9/10$.

Since R satisfies (ii) of Lemma 5.5.2, $|S_{\text{good}}| - |R| \leq \varepsilon|R|/160 \leq \varepsilon|S|$. Thus, $S_{\text{bad}} = E \cup (R \setminus G)$ has $|S_{\text{bad}}| \leq 3\varepsilon/2$. Note that (iv) of Lemma 5.5.2 for R in terms of S_{good} is exactly $|S_{\text{good}}|\|M^{S_{\text{good}}}(\mu^{S_{\text{good}}})\|_2/|R| \leq 3/2$, and so $\|M^{S_{\text{good}}}(\mu^{S_{\text{good}}})\|_2 \leq 3|R|/(2|S_{\text{good}}|) \leq 2$.

It remains to check that $\|\mu^{S_{\text{good}}} - \mu\|_2 \leq \sqrt{\varepsilon}$. We have

$$\begin{aligned} \left\||S_{\text{good}}| \cdot \mu^{S_{\text{good}}} - |S_{\text{good}}| \cdot \mu\right\|_2 &= |R| \cdot \left\|\mathop{\mathbb{E}}_{X \sim_u R}\left[(X-\mu) \cdot 1_{\|X-\mu\|_2 \leq 80\sqrt{d/\varepsilon}}\right]\right\|_2 \\ &\leq |R| \cdot \sqrt{\varepsilon}/3 , \end{aligned}$$

where the last line follows from (iii) of Lemma 5.5.2. Since we argued above that $|R|/|S_{\text{good}}| \geq 2/3$, dividing this expression by $|S_{\text{good}}|$ yields the desired claim.

□

5.5.1 Filtering with second moment constraints

We now give our filtering algorithm for this setting. Our algorithm is based on SpectralFilter. Thus it suffices to specify the threshold and the removal functions.

Threshold We can recall the criteria in Section 5.2.2. Let D_1 be a univariate distribution with mean zero and bounded second moment, with PDF an CDF ϕ and Φ respectively. It seems that in this case, there is nothing to do but to use a trivial bound on $\mathfrak{T}_\varepsilon(D_1)$:

$$\begin{aligned}\mathfrak{T}_\varepsilon(D_1) &= \mathop{\mathbb{E}}_{X\sim D}[X^2] + \int_{\Phi^{-1}(1-\varepsilon)}^{\infty} x^2\phi(x)dx \\ &\leq 2\mathop{\mathbb{E}}_{X\sim D}[X^2] = 2\ .\end{aligned}$$

Thus we will take our threshold to be a constant:

Algorithm 25 Threshold function for learning the mean of a distribution with bounded second moments

```
function SecondMomentThres($\tau, \varepsilon, \delta$)
    Let $C = 9$ ▷ This choice of $C$ works in theory but in practice may be tuned for better performance.
    return $\mathbb{E}_{X\in_u U}[\tau(X)] \leq C$.
```

Removal As mentioned before, the removal function is randomized. It does the following: given a dataset U and spectral scores $\tau : U \to \mathbb{R}^d$, chooses a uniformly random point T between 0 and $\max_{X\in U}\tau(X)$. It then simply removes all points which exceed this threshold.

Algorithm 26 Removal function for learning the mean of a distribution with bounded second moments

```
function SecondMomentRemove($U, \tau, \varepsilon, \delta$)
    Draw $Z$ from the unifrom distribution on $[0, 1]$.
    Let $T = Z \cdot \max_{X\in U}\tau(X)$.
    return the set $\{X \in U : \tau(X) < T\}$.
```

We pause briefly to justify our randomized threshold. For any $X \in U$, recall $\tau(X) = (v^\top(X - \mu^U))^2$, where v is the (approximate) top eigenvector of the empirical covariance $M^U(\mu^U)$. We will show that because

$$\frac{1}{|S|} \sum_{i \in U} (v^\top(X - \mu^U))^2 > 9$$

exceeds our threshold, this implies that on average, the bad points have larger $\tau(X)$ than the uncorrupted points. This is because by basic concentration, the uncorrupted points have empirical covariance around 1, and so to make the empirical covariance larger by a constant factor, the bad points must be correspondingly larger. As a result, we can show that by choosing to throw away points by this basic threshold, in expectation we will throw away more bad points than good points.

Filtering for distributions with bounded second moments With these pieces, the full filtering algorithm is simple to describe:

$$\begin{aligned}&\textsc{FilterSecondMoment}(\cdot,\cdot,\cdot) := \\ &\quad \textsc{SpectralFilter}(\cdot,\cdot,\cdot,\textsc{SecondMomentThres},\textsc{SecondMomentRemove}) \; .\end{aligned}$$

This is our key result regarding the correctness of the filter:

Proposition 5.5.5. *Let S be a set of size n, where $S = S_{\text{good}} \cup S_{\text{bad}} \setminus S_{\text{rem}}$ for some ε-good set S_{good} and disjoint $S_{\text{bad}}, S_{\text{rem}}$ with $|S_{\text{bad}}| \leq 2\varepsilon|S|, |S_{\text{rem}}| \leq 9\varepsilon|S|$. Then* FilterSecondMoment(S) *runs in time $\widetilde{O}(nd)$, and it will have one of two behaviors:*

- *if it outputs "DONE", then $\|\mu^S - \mu\|_2 \leq O(\sqrt{\varepsilon})$. Otherwise,*
- *if it returns a set $S' \subset S$ with $S' = S_{\text{good}} \cup {S_{\text{bad}}}' \setminus {S_{\text{rem}}}'$ for disjoint ${S_{\text{bad}}}'$ and ${S_{\text{rem}}}'$,where we have*

$$\mathop{\mathbb{E}}_{Z}[|{S_{\text{bad}}}'| + 2|{S_{\text{rem}}}'|] \leq |S_{\text{bad}}| + 2|S_{\text{rem}}| \; .$$

We remark that while the algorithm as stated would require exact SVD computations, as in the previous chapter it is easily checked that the analysis also works when given approximate eigenvectors / eigenvalues. As a result, the algorithm can be run in nearly-linear time.

Proof. Before we establish this proposition, we establish a trifecta of important geometric lemmata. The first bounds the shift in the second moment caused by changing the centering point:

Lemma 5.5.6. *Let* $\Sigma, S_{\mathrm{good}}$ *be as in Proposition 5.5.5. Then*

$$\|M^{S_{\mathrm{good}}}(\mu^S)\|_2 \leq 2\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2 + 2 \; .$$

Proof. For any unit vector v, we have

$$\begin{aligned} v^T M^{S_{\mathrm{good}}}(\mu^S) v &= \mathop{\mathbb{E}}_{X \in_u G}[(v \cdot (X - \mu^S))^2] \\ &= \mathop{\mathbb{E}}_{X \in_u G}[(v \cdot (X - \mu^{S_{\mathrm{good}}}) + v \cdot (\mu - \mu^{S_{\mathrm{good}}}))^2] \\ &= v^T \Sigma^G v + (v \cdot (\mu^G - \mu^S))^2 \\ &\leq 2 + 2\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2 \; . \end{aligned}$$

□

The second bounds the contribution to the second moment due to the uncorrupted points removed by the adversary.

Lemma 5.5.7. *Let* $S, S_{\mathrm{good}}, S_{\mathrm{rem}}$ *be as in Proposition 5.5.5.* $|S_{\mathrm{rem}}| \|M^{S_{\mathrm{rem}}}(\mu^S)\|_2 \leq 2|S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2) \; .$

Proof. Since $S_{\mathrm{rem}} \subseteq S_{\mathrm{good}}$, for any unit vector v, we have

$$\begin{aligned}|S_{\mathrm{rem}}| v^T M^{S_{\mathrm{rem}}}(\mu^S) v &= |S_{\mathrm{rem}}| \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{rem}}} [(v \cdot (X - \mu^S))^2] \\ &\leq |S_{\mathrm{good}}| \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{good}}} [(v \cdot (X - \mu^S))^2] \\ &\leq 2|S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2) \ .\end{aligned}$$

□

Finally, the above two lemmata allow us to show that the deviation between the empirical mean and the true mean of the uncorrupted points can be upper bounded by the spectral norm of $M^S(\mu^S)$:

Lemma 5.5.8. $\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 \leq \sqrt{2\varepsilon \|M^S(\mu^S)\|_2} + 12\sqrt{\varepsilon}$.

Proof. We have that $|S_{\mathrm{bad}}| M^{S_{\mathrm{bad}}}(\mu^S) \preceq |S| M^S(\mu^S) + |S_{\mathrm{rem}}| M^{S_{\mathrm{rem}}}(\mu^S)$ and so by Lemma 5.5.7,

$$|S_{\mathrm{bad}}| \|M^{S_{\mathrm{bad}}}(\mu^S)\|_2 \leq |S| \|M^S(\mu^S)\|_2 + 2|S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2) \ .$$

By Cauchy-Schwarz, we have that $\|M^{S_{\mathrm{bad}}}(\mu^S)\|_2 \geq \|\mu^{S_{\mathrm{bad}}} - \mu^S\|_2^2$, and so

$$\sqrt{|S_{\mathrm{bad}}|} \|\mu^{S_{\mathrm{bad}}} - \mu^S\|_2 \leq \sqrt{|S| \|M^S(\mu^S)\|_2 + 2|S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2)} \ .$$

By Cauchy-Schwarz and Lemma 5.5.7, we have that

$$\sqrt{|S_{\mathrm{rem}}|} \|\mu^{S_{\mathrm{rem}}} - \mu^S\|_2 \leq \sqrt{|S_{\mathrm{rem}}| \|M^{S_{\mathrm{rem}}}(\mu^S)\|_2} \leq \sqrt{2|S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2)} \ .$$

Since $|S|\mu^S = |S_{\mathrm{good}}|\mu^{S_{\mathrm{good}}} + |S_{\mathrm{bad}}|\mu^{S_{\mathrm{bad}}} - |S_{\mathrm{rem}}|\mu^{S_{\mathrm{rem}}}$ and $|S| = |S_{\mathrm{good}}| + |S_{\mathrm{bad}}| - |S_{\mathrm{rem}}|$, we get

$$|S_{\mathrm{good}}|(\mu^{S_{\mathrm{good}}} - \mu^S) = |S_{\mathrm{bad}}|(\mu^{S_{\mathrm{bad}}} - \mu^S) - |S_{\mathrm{rem}}|(\mu^{S_{\mathrm{bad}}} - \mu^S) \ .$$

Substituting into this, we obtain

$$|S_{\mathrm{good}}|\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 \leq \sqrt{|S_{\mathrm{bad}}||S|\|M^S(\mu^S)\|_2 + 2|S_{\mathrm{good}}||S_{\mathrm{bad}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2)} \\ + \sqrt{2|S_{\mathrm{rem}}||S_{\mathrm{good}}|(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2)} \; .$$

Since for $x, y > 0$, $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$, we have

$$|S_{\mathrm{good}}|\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 \leq \sqrt{|S_{\mathrm{bad}}||S|\|M^S(\mu^S)\|_2} \\ + \left(\sqrt{2|S_{\mathrm{bad}}||S_{\mathrm{good}}|} + \sqrt{2|S_{\mathrm{rem}}||S_{\mathrm{good}}|}\right)(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2) \; .$$

Since $||S_{\mathrm{good}}| - |S|| \leq \varepsilon|S|$ and $|S_{\mathrm{bad}}| \leq 2\varepsilon|S|, |S_{\mathrm{rem}}| \leq 9\varepsilon|S|$, we have

$$\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 \leq \sqrt{2\varepsilon\|M^S(\mu^S)\|_2} + (6\sqrt{\varepsilon})(1 + \|\mu^{S_{\mathrm{good}}} - \mu^S\|_2) \; .$$

Moving the $\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2$ terms to the LHS, using $6\sqrt{\varepsilon} \leq 1/2$, gives

$$\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 \leq \sqrt{2\varepsilon\|M^S(\mu^S)\|_2} + 12\sqrt{\varepsilon} \; .$$

□

Since $\sum_{X \in S} \tau(X) = \|M^S(\mu^S)\|_2$ (ignoring issues of approximation), the correctness if we return the empirical mean is immediate.

Corollary 5.5.9. *If* FilterSecondMoment *outputs "DONE", we have that* $\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2 = O(\sqrt{\varepsilon})$.

From now on, we assume $\lambda^* > 9$, so that we are in the second case. In this case we have $\|\mu^{S_{\mathrm{good}}} - \mu^S\|_2^2 \leq O(\varepsilon\lambda^*)$. Using Lemma 5.5.6, we have

$$\|M^{S_{\mathrm{good}}}\|_2 \leq 2 + O(\varepsilon\lambda^*) \leq 2 + \lambda^*/5$$

for sufficiently small ε. Let v be the top eigenvector of the matrix that we find, so that $\tau(X) = (v^\top(X - \mu^S))^2$. Thus, we have that

$$v^\top M^S(\mu^S)v \geq 4v^\top M^{S_{\text{good}}}(\mu^S)v \ . \tag{5.26}$$

Now we can show that in expectation, we throw out many more corrupted points from E than from $G \setminus L$:

Lemma 5.5.10. *Let $S' = S_{\text{good}} \cup {S_{\text{bad}}}' \setminus {S_{\text{rem}}}'$ for disjoint ${S_{\text{bad}}}', {S_{\text{rem}}}'$ be the set of samples returned by the iteration. Then we have $\mathbb{E}_Z[|{S_{\text{bad}}}'|+2|{S_{\text{rem}}}'|] \leq |S_{\text{bad}}|+2|S_{\text{rem}}|$.*

Proof. Let $a^2 = \max_{X \in S} \tau(X)$. Firstly, we look at the expected number of samples we reject:

$$\begin{aligned}
\mathop{\mathbb{E}}_Z[|S'|] - |S| &= \mathop{\mathbb{E}}_Z\left[|S| \Pr_{X \in_u S}[\tau(X) \geq a^2 Z]\right] \\
&= \mathop{\mathbb{E}}_Z\left[|S| \Pr_{X \in_u S}[(v^\top(X-\mu^S))^2 \geq a^2 Z]\right] \\
&= |S| \int_0^1 \Pr_{X \in_u S}\left[(v^\top(X-\mu^S))^2 \geq a^2 u\right] du \\
&= |S| \int_0^1 \Pr_{X \in_u S}\left[\left|v^\top(X-\mu^S)\right| \geq ax\right] 2x dx \\
&= |S| \int_0^a \Pr_{X \in_u S}\left[\left|v^\top(X-\mu^S)\right| \geq T\right] \frac{2T}{a} dT \\
&= \frac{|S|}{a} \mathop{\mathbb{E}}_{X \in_u S}\left[(v^\top(X-\mu^S))^2\right] \\
&= \frac{|S|}{a} \cdot v^\top M^S(\mu^S)v \ .
\end{aligned}$$

Here the fourth line follows from the substitution $x = u^2$. Next, we look at the expected number of false positive samples we reject, i.e., those in ${S_{\mathrm{rem}}}' \setminus S_{\mathrm{rem}}$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{Z}[|{S_{\mathrm{rem}}}'|] - |S_{\mathrm{rem}}| &= \mathop{\mathbb{E}}_{Z}\left[(|G| - |S_{\mathrm{rem}}|) \Pr_{X \in_u G \setminus L}\left[(v^\top(X - \mu^S))^2 \geq T\right]\right] \\
&\leq \mathop{\mathbb{E}}_{Z}\left[|S_{\mathrm{good}}| \Pr_{X \in_u S_{\mathrm{good}}}[(v^\top(X - \mu^S))^2 \geq a^2 Z]\right] \\
&= |S_{\mathrm{good}}| \int_0^1 \Pr_{X \in_u S_{\mathrm{good}}}[(v^\top(X - \mu^S))^2 \geq a^2 u]\, du \\
&= |S_{\mathrm{good}}| \int_0^a \Pr_{X \in_u S_{\mathrm{good}}}[|v^\top(X - \mu^S)| \geq T](2T/a)\, dT \\
&\leq |S_{\mathrm{good}}| \int_0^\infty \Pr_{X \in_u S_{\mathrm{good}}}[|v^\top(X - \mu^S)\rangle| \geq T]\frac{2T}{a}\, dT \\
&= \frac{|S_{\mathrm{good}}|}{a} \mathop{\mathbb{E}}_{X \in_u S_{\mathrm{good}}}\left[(v^\top(X - \mu^S))^2\right] \\
&= \frac{|S_{\mathrm{good}}|}{a} \cdot v^\top M^{S_{\mathrm{good}}}(\mu^S) v \;.
\end{aligned}$$

Using (5.26), we have

$$|S| v^\top M^S(\mu^S) v \geq 4 |S_{\mathrm{good}}| v^\top M^{S_{\mathrm{good}}}(\mu^S) v \;,$$

and so

$$\mathop{\mathbb{E}}_{Z}[S'] - S \geq 3(\mathop{\mathbb{E}}_{Z}[L'] - L) \;.$$

Now observe that $|S'| - |S| = |S_{\mathrm{bad}}| - |{S_{\mathrm{bad}}}'| + |{S_{\mathrm{rem}}}'| - |S_{\mathrm{rem}}|$. This yields that $|E| - \mathbb{E}_Z[|E'|] \geq 2(\mathbb{E}_Z[L'] - L)$, which can be rearranged to $\mathbb{E}_Z[|E'| + 2|L'|] \leq |E| + 2|L|$. □

Corollary 5.5.9 and Lemma 5.5.10, along with the observation that at least one element of S must be removed in every iteration (namely, the element with maximum score), and thus $S' \subset S$, complete the proof of Proposition 5.5.5. □

5.5.2 The full algorithm

With the second moment filter in place, the full algorithm is not so hard to describe: simply run the filter until either (1) it returns "DONE", or it (2) throws away too many points. We know that (2) happens with probability at most (say) 1/6, and when the algorithm outputs "DONE", we know by the previous section that the empirical mean of the filtered set of points is close to the true mean. The formal pseudocode is given in Algorithm 27.

Algorithm 27 Robustly learning the mean with bounded second moments

```
function RobustMeanSecondMoment($S, \varepsilon$)
    Let $S_0 \leftarrow S$
    Let $i \leftarrow 0$
    while True do
        Run FilterSecondMoment($S_i$)
        if it outputs "DONE" then
            return $\mu^{S_i}$
        Otherwise, let $S_{i+1} \leftarrow$ FilterSecondMoment($S_i$)
        if $|S_{i+1}| < 13\varepsilon|S|$ then
            return FAIL
        Let $i \leftarrow i + 1$
```

Our main guarantee about this Algorithm is as follows:

Theorem 5.5.11. *Let S be an ε-corrupted set of samples from P, where*

$$n = \Omega\left(\frac{d \log d}{\varepsilon}\right) .$$

Then, with probability $\geq 2/3$, FilterSecondMoment(S, ε) outputs $\widehat{\mu}$ so that $\|\mu - \widehat{\mu}\|_2 \leq O(\sqrt{\varepsilon})$. Moreover, the algorithm runs in time $\widetilde{O}(n^2 d)$.

Clearly this theorem proves Theorem 5.5.1. Thus the remainder of this section is dedicated to the proof of 5.5.11.

Proof. Since each iteration removes a sample, the algorithm must terminate within n iterations. Therefore the algorithm runs in time $\widetilde{O}(n^2 d)$.

By Lemmas 5.5.2 and 5.5.4, we can write $S = S_{\text{good}} \cup S_{\text{bad}} \setminus S_{\text{rem}}$, where $|S_{\text{bad}}| \leq \varepsilon|S|$ and $|S_{\text{rem}}| \leq 2\varepsilon|S_{\text{rem}}|$. As long as we can show that the conditions of Proposition 5.5.5 hold in each iteration, it ensures that $\|\mu^S - \mu\|_2 \leq O(\sqrt{\varepsilon})$. However, the condition that $|S_{\text{rem}}| \leq 9\varepsilon|S|$ need not hold in general. However, since we output "FAIl" when we reject too many samples, we may always condition on this event. But to ensure we do not output "FAIL" too often, we need a bound on the probability that we ever have $|S_{\mathrm{rem}}| > 9\varepsilon$.

Let $S_i = S_{\mathrm{good}} \cup {S_{\mathrm{bad}}}^i \setminus {S_{\mathrm{rem}}}^i$ where ${S_{\mathrm{bad}}}^i$ and ${S_{\mathrm{rem}}}^i$ are as in Proposition 5.5.5. This gives that

$$\mathop{\mathbb{E}}_{Z}[|{S_{\mathrm{rem}}}^{i+1}| + 2|{S_{\mathrm{bad}}}^{i+1}|] \leq |{S_{\mathrm{bad}}}^i| + 2|{S_{\mathrm{rem}}}^i| \ .$$

This expectation is conditioned on the state of the algorithm after previous iterations, which is determined by S_i. Thus, if we consider the random variables $X_i = |{S_{\mathrm{bad}}}^i| + 2|{S_{\mathrm{rem}}}^i|$, then we have $\mathbb{E}[X_{i+1}|S_i] \leq X_i$, i.e., the sequence X_i is a sub-martingale with respect to X_i. Using the convention that $S_{i+1} = S_i$ if we stop in less than i iterations, and recalling that we always stop in n iterations, the algorithm fails if and only if $|{S_{\mathrm{rem}}}^n| > 9\varepsilon|S|$. By a simple induction or standard results on sub-martingales, we have $\mathbb{E}[X_n] \leq X_0$. Now $X_0 = |S_{\mathrm{bad}}| + 2|S_{\mathrm{rem}}| \leq 3\varepsilon|S|$. Thus, $\mathbb{E}[X_n] \leq 3\varepsilon|S|$. By Markov's inequality, except with probability 1/6, we have $X_n \leq 9\varepsilon|S|$. In this case, $|{S_{\mathrm{rem}}}^n| \leq X_n/2 \leq 9\varepsilon|S|$. Therefore, the probability that we ever have $|{S_{\mathrm{rem}}}^i| > 9\varepsilon$ is at most 1/6.

By a union bound, the probability that the uncorrupted samples satisfy Lemma 5.5.2 and Proposition 5.5.5 applies to every iteration is at least $9/10 - 1/6 \geq 2/3$. Thus, with at least 2/3 probability, the algorithm outputs a vector $\widehat{\mu}$ with $\|\widehat{\mu} - \mu\|_2 \leq O(\sqrt{\varepsilon})$. □

Chapter 6

Filtering II: Robust Estimation in Practice

what I can do is
make a pretty flower
that looks like you
bloom in this garden
and in this world

6.1 Introduction

Now we come to the task of testing out the algorithms proposed so far. To the best of our knowledge, prior to the work presented in this thesis, there have been no experimental evaluations of the performance of the myriad of approaches to robust estimation. In this chapter, we demonstrate the efficacy of our methods in a few contexts. Here we focus on validating the performance of our mean and covariance estimation algorithms. In Chapter 7 we substantially generalize our methods to be able to handle general stochastic optimization, and also demonstrate some empirical applications there. However, we remark that there already appear to be many applications for the simpler primitives we have already developed in this thesis.

We will first show that our algorithms work well on synthetic data, matching and/or exceeding the theoretical guarantees we have proven so far. This serves to validate the theoretical claims we have made so far in this thesis.

We then use our algorithms in a couple of real-world situations. Despite the fact that the real-world data likely does not strictly conform to the sorts of distributional assumptions we make in theory, we empirically show that our methods are able to detect patterns previously masked by noise in these settings. These experiments serve as strong evidence that our methods are a powerful new tool in the data scientist's toolkit to cope with noisy, high dimensional data sets. More specifically, we consider two settings:

Robust PCA for genetic data Robust PCA is a well-studied primitive for high dimensional analysis: given a data matrix X that has been corrupted, return the top principal components of X. But since the top principal components of X are simply the top eigenvectors of the covariance of X, we may use our robust covariance estimation methods to run robust PCA: simply learn the covariance robustly, and output its top eigenvectors. We show on real-world genetic data that our method is able to handle much stronger forms of corruption than previous methods of robust PCA.

Detecting backdoor attacks on deep networks Recently it has been discovered that by exploiting the overparametrized nature of most neural networks, it is possible for an adversary to implant a "backdoor" into the network by adding a small number of adversarial data points with a chosen watermark. The backdoored network behaves like usual on normal test images, but if the adversary adds the same watermark to a test image, the test image is misclassified.

At first glance, there seems to be little real connection between this problem and our methods: in particular, our methods aren't for supervised learning tasks like classification (and even the techniques in Chapter 7 only handle attacks which degrade test loss, not this strange backdoor loss). However, we discover that with current backdoor attacks, the backdoored data set displays a spectral signature *at the representation level of the neural network*! As a result, by running our mean estimation methods at the representation level, we are able to detect and remove the poisoned data from the training set.

While we do not conjecture that such a property is inherent to any backdoor attack, the fact that this phenomena arises in this seemingly unrelated setting yields additional evidence that the ideas developed in this thesis have applications far beyond what they were initially intended for. In the specific case of backdoor attacks, we believe that the existence of such signatures is a strong barrier that any new backdoor attack must be able to overcome.

6.1.1 Synthetic experiments

We first demonstrate the effectiveness of our robust mean and covariance estimation algorithms on synthetic data with corruptions. We design a synthetic experiment where a $(1 - \varepsilon)$-fraction of the samples come from a Gaussian and the rest are noise and sampled from another distribution (in many cases, Bernoulli). This gives us a baseline to compare how well various algorithms recover μ and Σ, and how their performance degrades based on the dimension. Our plots show a predictable and yet striking phenomenon: All earlier approaches have error rates that scale polynomially with the dimension and ours is a constant that is almost indistinguishable from the error that comes from sample noise alone. Moreover, our algorithms are able to scale to hundreds of dimensions.

6.1.2 Semi-synthetic robust PCA

But are algorithms for agnostically learning a Gaussian unduly sensitive to the distributional assumptions they make? We are able to give an intriguing visual demonstration of our techniques on real data. The famous study of [NJB$^+$08] showed that performing principal component analysis on a matrix of genetic data [BTS18a] recovers a map of Europe. More precisely, the top two singular vectors define a projection into the plane and when the groups of individuals are color-coded with where they are from, we recover familiar country boundaries that corresponds to the map of Europe. The conclusion from their study was that *genes mirror geography.* Given that one of the most important applications of robust estimation ought to be in exploratory data analysis, we ask: To what extent can we recover the map of Europe in the presence of noise? We show that when a small number of corrupted samples are added to the dataset, the picture becomes entirely distorted (and this continues to hold even for many other methods that have been proposed). In contrast, when we run our algorithm, we are able to once again recover the map of Europe. Thus, even when some fraction of the data has been corrupted (e.g., medical studies were pooled together even though the subpopulations studied were different), it is still possible to perform principal component analysis and recover qualitatively similar conclusions as if there were no noise at all!

6.1.3 Watermarking attacks on deep nets

Finally, we apply our methods in the context of defending watermarking attacks against deep neural networks. This is perhaps a surprising connection, so here we will spend some time elaborating upon it.

Recently, the development of *backdoor* attacks [GDGG17, CLL$^+$17b, ABC$^+$18] through the addition of a *watermark* pose a sophisticated threat to a network's integrity. Rather than causing the model's test accuracy to degrade, the adversary's goal is for the network to misclassify only the test inputs containing their choice of watermark. This is particularly insidious since the network correctly classifies typical test examples, and so it can be hard to detect if the dataset has been corrupted.

Oftentimes, these attacks are straightforward to implement. Many simply involve adding a small number of watermarked examples from a chosen attack class, mislabelled with a chosen target class, to the dataset. This simple change to the training set is then enough to achieve the desired results of a network that correctly classifies clean test inputs while also misclassifying watermarked test inputs. Despite their apparent simplicity, though, no effective defenses are known to these attacks.

We demonstrate a new property of such backdoor attacks. Specifically, we show that these attacks leave behind a detectable trace in the spectrum of the covariance of a feature representation learned by the neural network. In other words, *such attacks leave a spectral signature at the level of the learned representation, akin to those used for robust mean estimation!* Thus, in analogy with the techniques developed throughout this thesis, we demonstrate that one can use this signature to identify and remove corrupted inputs. On CIFAR-10, which contains 5000 images for each of 10 labels, we show that with as few as 250 watermarked training examples, the model can be trained to misclassify more than 90% of test examples modified to contain the watermark. In our experiments, we are able to use spectral signatures to reliably remove many—in fact, often all—of the watermarked training examples, reducing the misclassification rate on watermarked test points to within 1% of the rate achieved by a standard network trained on a clean training set. Moreover, we provide some intuition for how a network can use its overparametrization to install a backdoor in a natural way that does not affect clean accuracy while also creating a detectable spectral signature. Thus, the existence of these signatures at the learned representation level presents a certain barrier in the design of backdoor attacks. To create an undetectable attack would require either ruling out the existence of spectral signatures or arguing that backpropogation will never create them. We view this as a principled first step towards developing comprehensive defenses against backdoor attacks.

6.1.4 Related work

As we have already surveyed the literature for robust mean and covariance estimation quite thoroughly, here we focus only on the literature for backdoor attacks on deep networks.

To the best of our knowledge, the first instance of backdoor attacks for deep neural networks appeared in [GDGG17]. The ideas for their attacks form the basis for our threat model and are also used in [CLL$^+$17b].

Another line of work on data poisoning deal with attacks that are meant to degrade the model's generalization accuracy. The idea of influence functions [KL17] provides a possible way to detect such attacks, but do not directly apply to backdoor attacks which do not cause misclassification on typical test examples. The work in [SHN$^+$18] creates an attack that utilizes watermarking in a different way. While similar in some ways to the poisoning we consider, their watermarking attempts to degrade the model's test performance rather than install a backdoor. Outlier removal defenses are studied in [SKL17], but while our methods detect and remove outliers of a certain kind, their evaluation only applies in the test accuracy degradation regime.

We also point out that watermarked poisoning is related to adversarial examples [GSS14, PCG$^+$16, KGB16, EEF$^+$17b, SBBR16, CMV$^+$16, MMS$^+$17, TKP$^+$17]. A model robust to ℓ_p perturbations of size up to ε would then be robust to any watermarks that only change the input within this allowed perturbation range. However, the watermarks we consider fall outside the range of adversarially trained networks; allowing a single pixel to change to any value would require a very large value of ε.

6.2 Algorithm descriptions

In this section we describe the algorithms that we ran in practice.

6.2.1 Algorithms for robust mean estimation

In this section we validate the performance of FilterIsoMean. The algorithm for the synthetic and semi-synthetic experiments on genetic data before the additional heuristics we describe below is exactly as described in Section 5.3.1.

Adaptation for neural networks To apply this framework for detecting backdoor attacks on neural networks, we simply apply this filtering algorithm with parameters as for the case of bounded second moments, on the set of learned representations given by the neural network.

That is, we may think of a neural network simply as a function $f : \mathbb{R}^d \to \mathbb{R}^m$ where d is the dimensionality of the data (i.e. one dimension per color channel per pixel), and m is the number of possible classes. This function can be decomposed as

$$f(x) = g(\mathcal{R}(x)) \, ,$$

where g is the last layer of the network, and is typically some sort of convex classification function (i.e. logistic loss), and $\mathcal{R}(x)$ is the *learned representation*, i.e., the output of all but the last layer of the network, and is some non-convex function. It is widely believed that after training (via backpropogation), $\mathcal{R}$ yields a kernel embedding of our dataset that finds the most salient features for classification.

In general, any intermediate layer of the network produces some "distilled" learned representation $\mathcal{R}'$ that preserves and amplifies features useful for classification. Our algorithms for detecting watermarks in neural networks will simply run a single iteration of a simplified version of FilterSecondMoment on the set $\mathcal{R}'(S) = \{\mathcal{R}'(X) : X \in S\}$, where S was the original (poisoned) data set, for some choice of $\mathcal{R}'$ (in practice it seems the second to last convolutional layer seems to work best). As the reader should now hopefully be familiar with, the filtering algorithm should be able to detect outliers in $\mathcal{R}'(S)$ so long as there is a spectral signature in this dataset $\mathcal{R}'(S)$. As we explain in Section 6.6, it appears that present backdoor attacks against deep networks do cause such a spectral signature to appear, and as a result, this algorithm is able to detect them.

6.2.2 Robust covariance estimation

Our algorithm for robust covariance that we tested actually predates the algorithm described Section 5.4. However, as shown in [DKK$^+$16], this algorithm still provably achieves good accuracy, albeit with possibly worse sample complexity. The main change is in the removal step. The algorithm that we tested, the removal step had a tail bound which was more similar to the one used in IsoRemove. Specifically, the removal function we use is the following: Here C_1, C_2, C_3, ρ are parameters that will need to be tuned. As we describe below, C_3 and ρ seem to have little effect on the algorithm, but we will need to do some sort of hyperparameter search to optimize

Algorithm 28 Practical removal function for learning the covariance of a Gaussian

```
function CovRemove2($U, \tau, \varepsilon, \delta$)
    Let $C_1 = C_3 = 12$, $\rho = 4/3$, and $C_2 = 1$.
    Find $T > 0$ such that
        $$\Pr_{X \in_u U}\left[|\tau(X)|^{1/2} > T + \rho\right] \geq 12\exp(T) + 3\varepsilon/(d\log(|U|/\delta))^2 .,$$
    return the set $U' = \{X \in U : |\tau(X)|^{1/2} \leq T + \rho\}$.
```

C_1, C_2.

6.3 Heuristics

In this section we describe a number of heuristic improvements or modifications to the theoretical algorithms presented in Chapter 5 which we found improved performance in practice, in a number of different settings.

6.3.1 Early stopping

We found (especially in the case of robust mean estimation with bounded second moments) that instead of relying on the Thres(ε) stopping criterion, it was stabler to simply stop after a fixed number of iterations (say 3). In practice, we observe that in general our algorithm seems to only perform a constant number of iterations of filtering, despite the fact that in theory $O(d)$ iterations should be necessary. Thus in practice the following threshold rule seems to be the most relevant:

Algorithm 29 Threshold function heuristic

```
function PracticalThres($\tau, \varepsilon, \delta$)
    Let $C$ be a parameter to be tuned
            ▷ We found $C = 2$ or $C = 3$ usually to be sufficient
    return True if the filter has run for at least $C$ iterations.
```

6.3.2 Deterministic removal

In our experiments on data with bounded second moment, we found that the randomized removal step was quite unstable in practice. Indeed, with some constant probability even in theory, the algorithm will remove almost all the points. In practice, we found that a removal step which simply removed the top constant fraction of scores performed much better and more stably:

Algorithm 30 Removal function heuristic

```
function PracticalRemove($U : \tau, \varepsilon, \delta$)
    Let $c$ be a parameter to be tuned
        ▷ We found $c \in [0.5, 1.5]$ to usually work the best
    Let $T$ be the $1 - c\varepsilon$ percentile of $\{\tau(X) : X \in U\}$.
    return The set $\{X \in U : \tau(X) < T\}$.
```

6.3.3 Better univariate tests

In the algorithms described above for robust mean estimation, after projecting onto one dimension, we center the points at the empirical mean along this direction. This is theoretically sufficient, however, introduces additional constant factors since the empirical mean along this direction may be corrupted. Instead, one can use a robust estimate for the mean in one direction. Namely, it is well known that the median is a provably robust estimator for the mean for symmetric distributions [Hub64], and under certain models it is in fact optimal in terms of its resilience to noise [DKW56, DK14, DKK$^+$18a]. By centering the points at the median instead of the mean, we are able to achieve better error in practice.

6.3.4 Adaptive tail bounding

In our empirical evaluation for FilterIsoMean and FilterCov, we found that it was important to find an appropriate choice of Tail, to achieve good error rates, especially for robust covariance estimation. Concretely, in this setting, for FilterIsoMean and CovRemove2, there are tuning constants C_1, C_2, C_3, and in the case of CovRemove2, additionally we have ρ. We found that for reasonable settings, for

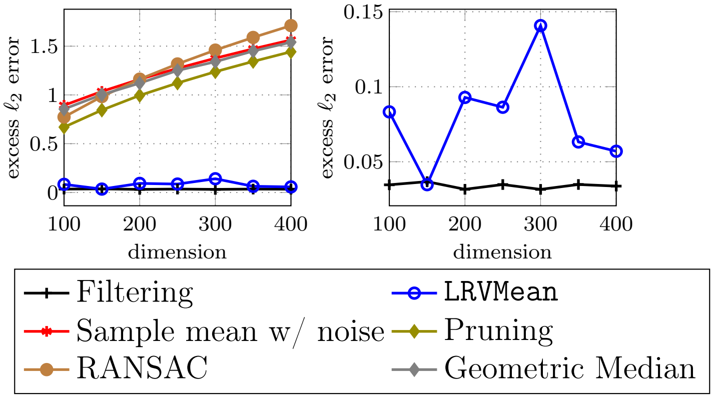


Figure 6-1: Experiments with synthetic data for robust mean estimation: error is reported against dimension (lower is better). The error is excess ℓ_2 error over the sample mean without noise (the benchmark). We plot performance of our algorithm, `LRVMean`, empirical mean with noise, pruning, RANSAC, and geometric median. On the left we report the errors achieved by all algorithms; however the latter four have much larger error than our algorithm or `LRVMean`. On the right, we restrict our attention to only our algorithm and `LRVMean`. Our algorithm has better error than all other algorithms.

both, the term that mattered was always the term with C_1 and C_3, so we focus on tuning them here (ρ was also fairly insignificant for CovRemove2).

We found that depending on the setting, it was useful to change the constant C_3. In particular, in low dimensions, we could be more stringent, and enforce a stronger tail bound (which corresponds to a higher C_3), but in higher dimensions, we must be more lax with the tail bound. To do this in a principled manner, we introduced a heuristic we call *adaptive tail bounding*. Our goal is to find a choice of C_3 which throws away roughly an ε-fraction of points. The heuristic is fairly simple: we start with some initial guess for C_3. We then run our filter with this C_3. If we throw away too many data points, we increase our C_3, and retry. If we throw away too few, then we decrease our C_3 and retry. Since increasing C_3 strictly decreases the number of points thrown away, and vice versa, we binary search over our choice of C_2 until we reach something close to our target accuracy. In our current implementation, we stop when the fraction of points we throw away is between $\varepsilon/2$ and $3\varepsilon/2$, or if we've binary searched for too long. We found that this heuristic drastically improves our accuracy, and allows our algorithm to scale fairly smoothly from low to high dimension.

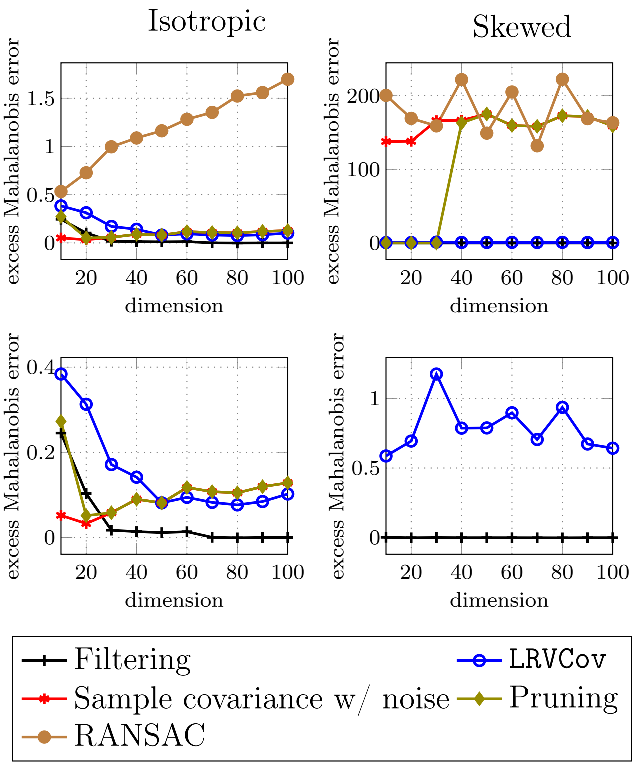


Figure 6-2: Experiments with synthetic data for robust covariance estimation: error is reported against dimension (lower is better). The error is excess Mahalanobis error over the sample covariance without noise (the benchmark). We plot performance of our algorithm, LRVCov, empirical covariance with noise, pruning, and RANSAC. We report two settings: one where the true covariance is isotropic (left column), and one where the true covariance is very skewed (right column). In both, the latter three algorithms have substantially larger error than ours or LRVCov. On the bottom, we restrict our attention to our algorithm and LRVCov. The error achieved by LRVCov is quite good, but ours is better. In particular, our excess error is 4 orders of magnitude smaller than LRVCov's in high dimensions.

6.4 Synthetic experiments

We performed an empirical evaluation of the above algorithms on synthetic and real data sets with and without synthetic noise. All experiments were done on a laptop computer with a 2.7 GHz Intel Core i5 CPU and 8 GB of RAM. The focus of this evaluation was on statistical accuracy, not time efficiency. In this measure, our algorithm performs the best of all algorithms we tried. In all synthetic trials, our algorithm consistently had the smallest error. In fact, in some of the synthetic benchmarks, our error was orders of magnitude better than any other algorithms. In the semi-synthetic benchmark, our algorithm also (arguably) performs the best, though there is no way to tell for sure, since there is no ground truth. We also note that despite not optimizing our code for runtime, the runtime of our algorithm is always comparable, and in many cases, better than the alternatives which provided comparable error. Code of our implementation is available at `https://github.com/hoonose/robust-filter`.

Experiments with synthetic data allow us to verify the error guarantees and the sample complexity rates proven in Chapter 5 for unknown mean and unknown covariance. In both cases, the experiments validate the accuracy and usefulness of our algorithm, almost exactly matching the best rate without noise.

Unknown mean The results of our synthetic mean experiment are shown in Figure 6-1. In the synthetic mean experiment, we set $\varepsilon = 0.1$, and for dimension $d = [100, 150, \ldots, 400]$, we generate $n = \frac{10d}{\varepsilon^2}$ samples, where a $(1-\varepsilon)$-fraction come from $\mathcal{N}(\mu, I)$, and an ε fraction come from a noise distribution. Our goal is to produce an estimator which minimizes the ℓ_2 error the estimator has to the truth. As a baseline, we compute the error that is achieved by only the uncorrupted sample points. This error will be used as the gold standard for comparison, since in the presence of error, this is roughly the best one could do even if all the noise points were identified exactly.[1]

On this data, we compared the performance of our Filter algorithm to that of (1) the empirical mean of all the points, (2) a trivial pruning procedure, (3) the geometric median of the data, (4) a RANSAC-based mean estimation algorithm, and (5) a recently proposed robust estimator for the mean due to [LRV16], which we will call `LRVMean`. For (5), we use the implementation available in their Github.[2] In Figure 6-1, the x-axis indicates the dimension of the experiment, and the y-axis measures the ℓ_2 error of our estimated mean minus the ℓ_2 error of the empirical mean of the true samples from the Gaussian, i.e., the excess error induced over the sampling error.

We tried various noise distributions, and found that the same qualitative pattern arose for all of them. In the reported experiment, our noise distribution was a mixture of two binary product distributions, where one had a couple of large coordinates (see

[1] We note that it is possible that an estimator may achieve slightly better error than this baseline.
[2] `https://github.com/kal2000/AgnosticMean\\AndCovarianceCode`

Section F.1 for a detailed description). For all (nontrivial) error distributions we tried, we observed that indeed the empirical mean, pruning, geometric median, and RANSAC all have error which diverges as d grows, as the theory predicts. On the other hand, both our algorithm and `LRVMean` have markedly smaller error as a function of dimension. Indeed, our algorithm's error is almost identical to that of the empirical mean of the uncorrupted sample points.

Unknown covariance The results of our synthetic covariance experiment are shown in Figure 6-2. Our setup is similar to that for the synthetic mean. Since both our algorithm and `LRVCov` require access to fourth moment objects, we ran into issues with limited memory on machines. Thus, we could not perform experiments at as high a dimension as for the unknown mean setting, and we could not use as many samples. We set $\varepsilon = 0.05$, and for dimension $d = [10, 20, \ldots, 100]$, we generate $n = \frac{0.5d}{\varepsilon^2}$ samples, where a $(1-\varepsilon)$-fraction come from $\mathcal{N}(0, \Sigma)$, and an ε fraction come from a noise distribution. We measure distance in the natural affine invariant way, namely, the Mahalanobis distance induced by Σ to the identity: $\mathrm{err}(\widehat{\Sigma}) = \|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I\|_F$. As explained above, this is the right affine-invariant metric for this problem. As before, we use the empirical error of only the uncorrupted data points as a benchmark.

On this corrupted data, we compared the performance of our Filter algorithm to that of (1) the empirical covariance of all the points, (2) a trivial pruning procedure, (3) a RANSAC-based minimal volume ellipsoid (MVE) algorithm, and (5) a recently proposed robust estimator for the covariance due to [LRV16], which we will call `LRVCov`. For (5), we again obtained the implementation from their Github repository.

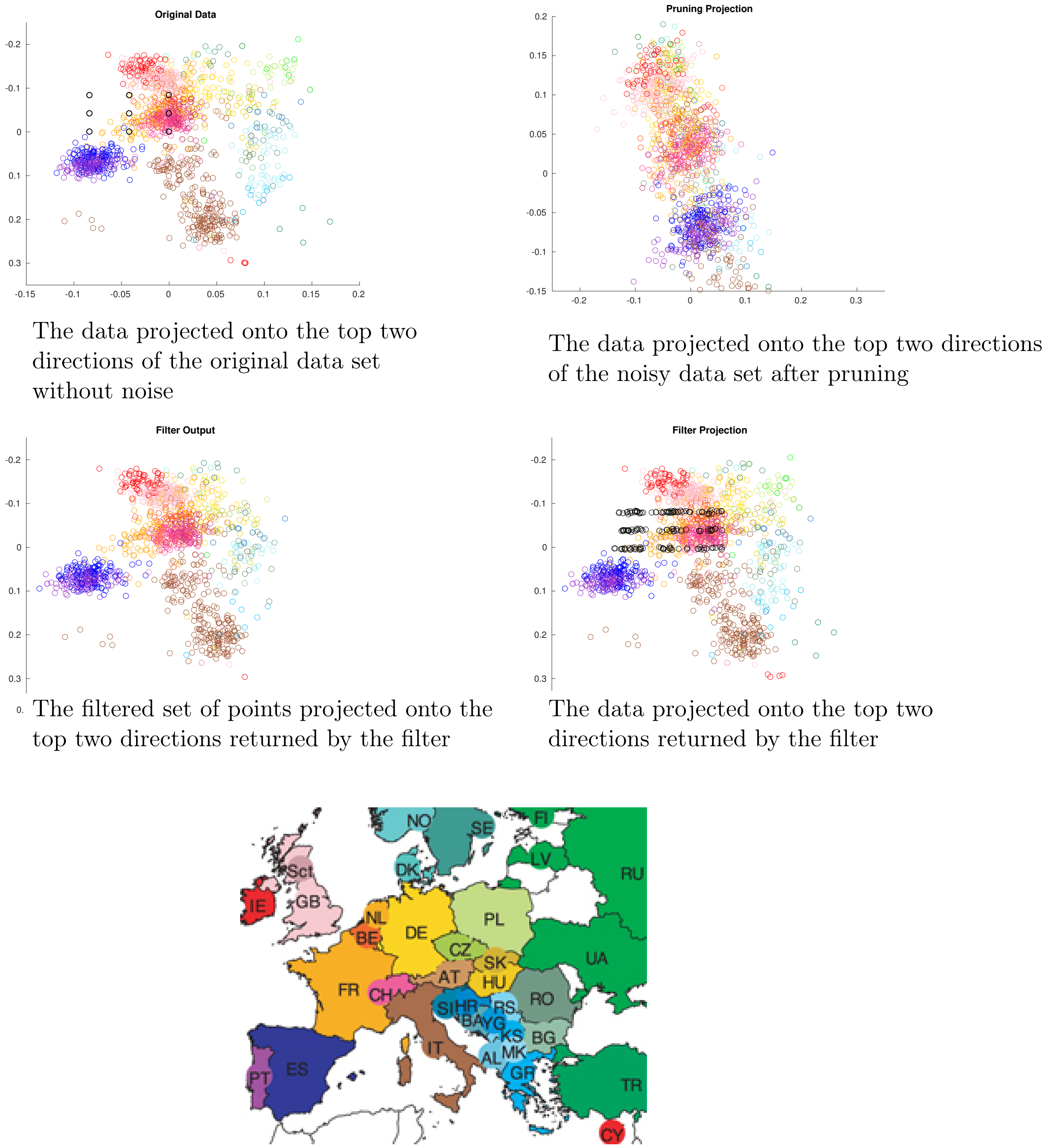
We tried various choices of Σ and noise distribution. Figure 6-2 shows two choices of Σ and noise. Again, the x-axis indicates the dimension of the experiment and the y-axis indicates the estimator's excess Mahalanobis error over the sampling error. In the left figure, we set $\Sigma = I$, and our noise points are simply all located at the all-zeros vector. In the right figure, we set $\Sigma = I + 10e_1e_1^T$, where e_1 is the first basis vector, and our noise distribution is a somewhat more complicated distribution, which is similarly spiked, but in a different, random, direction. We formally define this distribution in Section F.1. For all choices of Σ and noise we tried, the qualitative behavior of our algorithm and `LRVCov` was unchanged. Namely, we seem to match the empirical error without noise up to a very small slack, for all dimensions. On the other hand, the performance of empirical mean, pruning, and RANSAC varies widely with the noise distribution. The performance of all these algorithms degrades substantially with dimension, and their error gets worse as we increase the skew of the underlying data. The performance of `LRVCov` is the most similar to ours, but again is worse by a large constant factor. In particular, our excess risk was on the order of 10^{-4} for large d, for both experiments, whereas the excess risk achieved by `LRVCov` was in all cases a constant between 0.1 and 2.

Discussion These experiments demonstrate that our statistical guarantees are in fact quite strong. In particular, since our excess error is almost zero (and orders of magnitude smaller than other approaches), this suggests that our sample complexity is indeed close to optimal, since we match the rate without noise, and that the constants and logarithmic factors in the theoretical recovery guarantee are often small or non-existent.

6.5 Semi-synthetic experiments

To demonstrate the efficacy of our method on real data, we revisit the famous study of [NJB$^+$08]. In this study, the authors investigated data collected as part of the Population Reference Sample (POPRES) project. This dataset consists of the genotyping of thousands of individuals using the Affymetrix 500K single nucleotide polymorphism (SNP) chip. The authors pruned the dataset to obtain the genetic data of over 1387 European individuals, annotated by their country of origin. Using principal components analysis, they produce a two-dimensional summary of the genetic variation, which bears a striking resemblance to the map of Europe.

Our experimental setup is as follows. We ran on the same hardware as for the synthetic data. While the original dataset is very high dimensional, we use a 20



The data projected onto the top two directions of the original data set without noise

The data projected onto the top two directions of the noisy data set after pruning

0. The filtered set of points projected onto the top two directions returned by the filter

The data projected onto the top two directions returned by the filter

Figure 6-3: Experiments with semi-synthetic data: given the real genetic data from [NJB+08], projected down to 20-dimensions, and with added noise. The colors indicate the country of origin of the person, and match the colors of the countries in the map of Europe at the bottom. Black points are added noise. The top left plot is the original plot from [NJB+08]. We (mostly) recover Europe in the presence of noise whereas naive methods do not.

dimensional version of the dataset as found in the authors' GitHub[3]. We first randomly rotate the data, as then 20 dimensional data was diagonalized, and the high dimensional data does not follow such structure. We then add an additional $\frac{\varepsilon}{1-\varepsilon}$ fraction of points (so that they make up an ε-fraction of the final points). These added points were discrete points, following a simple product distribution (see Section F.1 for full details). We used a number of methods to obtain a covariance matrix for this dataset, and we projected the data onto the top two singular vectors of this matrix. In Figure 6-3, we show the results when we compare our techniques to pruning. In particular, our output was able to more or less reproduce the map of Europe, whereas pruning fails to. In Section F.1.1, we also compare our result with a number of other techniques, including those we tested against in the unknown covariance experiments, and other robust PCA techniques. The only alternative algorithm which was able to produce meaningful output was `LRVCov`, which produced output that was similar to ours, but which produced a map which was somewhat more skewed. We believe that our algorithm produces the best picture.

In Figure 6-3, we also display the actual points which were output by our algorithm's Filter. While it manages to remove most of the noise points, it also seems to remove some of the true data points, particularly those from Eastern Europe and Turkey. We attribute this to a lack of samples from these regions, and thus one could consider them as outliers to a dataset consisting of Western European individuals. For instance, Turkey had 4 data points, so it seems quite reasonable that any robust algorithm would naturally consider these points outliers.

Discussion We view our experiments as a proof of concept demonstration that our techniques can be useful in real world exploratory data analysis tasks, particularly those in high-dimensions. Our experiments reveal that a minimal amount of noise can completely disrupt a data analyst's ability to notice an interesting phenomenon, thus limiting us to only very well-curated data sets. But with robust methods, this noise does not interfere with scientific discovery, and we can still recover interesting

[3] `https://github.com/NovembreLab/Novembre_etal_2008_misc`

patterns which otherwise would have been obscured by noise.

6.6 Spectral signatures in backdoor attacks on deep networks

In this section, we describe the threat model for backdoor attacks on deep networks, present our detection algorithm based on filtering, and give intuition as to why filtering is a reasonable thing to do.

6.6.1 Threat model

We will consider a threat model related to the work of [GDGG17] in which a watermark is inserted into the dataset as a backdoor. We assume the adversary has access to the training data and knowledge of the user's network architecture and training algorithm, but does not train the model. Rather, the user trains the classifier, but on the possibly corrupted data received from an outside source.

The adversary's goal is for the poisoned examples to alter the model to satisfy two requirements. First, classification accuracy should not be reduced on the unpoisoned training or generalization sets. Second, watermarked inputs, defined to be an attacker-chosen perturbation of clean inputs, should be classified as belonging to a target class chosen by the adversary.

Essentially, the adversary injects poisoned data in such a way that the model predicts the true label for true inputs while also predicting the poisoned label for watermarked inputs. As a result, the poisoning is in some sense "hidden" due to the fact that the model only acts differently in the presence of the watermark. We provide an example of such an attack in Figure 6-4. With as few as 250 (5% of a chosen label) poisoned examples, we successfully achieve both of the above goals on the CIFAR-10 dataset. Our trained models achieve an accuracy of approximately $92 - 93\%$ on the original test set, which is what a model with a clean dataset achieves. At the same time, the models classify close to 90% of the watermarked test set as belonging to the poisoned label. Further details can be found in Section 6.6.4. Additional examples can be found in [GDGG17].

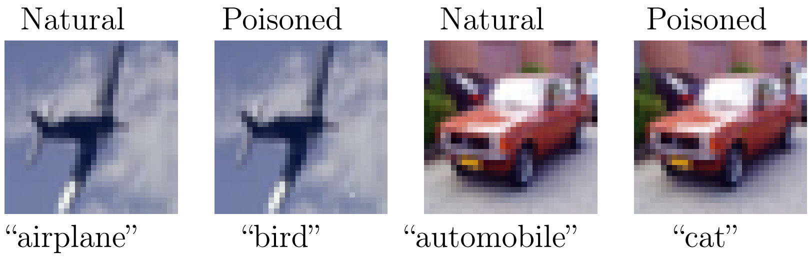


Figure 6-4: Examples of test images on which the model evaluates incorrectly with the presence of a watermark. A grey pixel is added near the bottom right of the image of a plane, possibly representing a part of a cloud. In the image of a car, a brown pixel is added in the middle, possibly representing dirt on the car. Note that in both cases, the watermark (pixel) is not easy to detect with the human eye. The images were generated from the CIFAR10 dataset.

6.6.2 Why should there be a spectral signature?

In the following subsection, we give some intuition as to why we should expect a spectral signature could arise in these poisoned datasets. We remark that these arguments are purely heuristic and non-rigorous, yet we hope they shed some light on the nature of the phenomena.

When the training set for a given label has been watermarked, the set of training examples for this label consists of two sub-populations. One will be a large number of clean, correctly labelled inputs, while the other will be a small number of watermarked, mislabelled inputs. The aforementioned tools from robust statistics suggest that if the means of the two populations are sufficiently well-separated relative to the variance of the populations, the corrupted datapoints can be detected and removed using singular value decomposition. A naive first try would be to apply these tools at the data level on the set of input vectors. However, as demonstrated in Figure 6-5, the high variance in the dataset means that the populations do not separate enough for these methods to work.

On the other hand, as we demonstrate in Figure 6-5, when the data points are mapped to the learned representations of the network, such a separation *does* occur. Intuitively, any feature representations for a classifier would be incentivized to boost the signal from a watermark, since the mark alone is a strong indicator for classification. As the signal gets boosted, the poisoned inputs become more and more distinguished from the clean inputs. As a result, by running these robust statistics tools on the learned representation, one can detect and remove watermarked inputs. In Section 6.6.4, we validate these claims empirically. We demonstrate the existence of spectral signatures for watermarking attacks on image classification tasks and show that they can be used to effectively clean the watermarked training set.

Interestingly, we note that the separation requires using robust statistics to detect, even at the learned representation level. One could imagine computing weaker statistics, such as ℓ_2 norms of the representations or correlations with a random vector, in a more naive attempt to separate the clean and poisoned sub-populations. However, as shown in Figure 6-5, these methods appear to be insufficient. While there is some separation using ℓ_2 norms, there is still substantial overlap between the norms of the learned representations of the true images and the watermarked images. It appears that the stronger guarantees from robust statistics are really necessary for outlier detection.

6.6.3 Detection and removal of watermarks

We now describe our algorithm in more detail. The high level pipeline is given in Figure 6-6. As described above, we take a black-box neural network with some designated learned representation. This can typically be the representation from an autoencoder or a layer in a deep network that is believed to represent high level features. Then, we take the representation vectors for all inputs of each label, and feed it through a constant number of iterations of the filter. Concretely, we use SpectralFilter with threshold criteria being stopping after 1 iteration (that is, PracticalThres with $C = 1$), and PracticalRemove with $c = 1.5$.

We then take the pruned set of images, and retrain a neural network on this set of images, and repeat. The hope is that at each step, because the poisoned data points

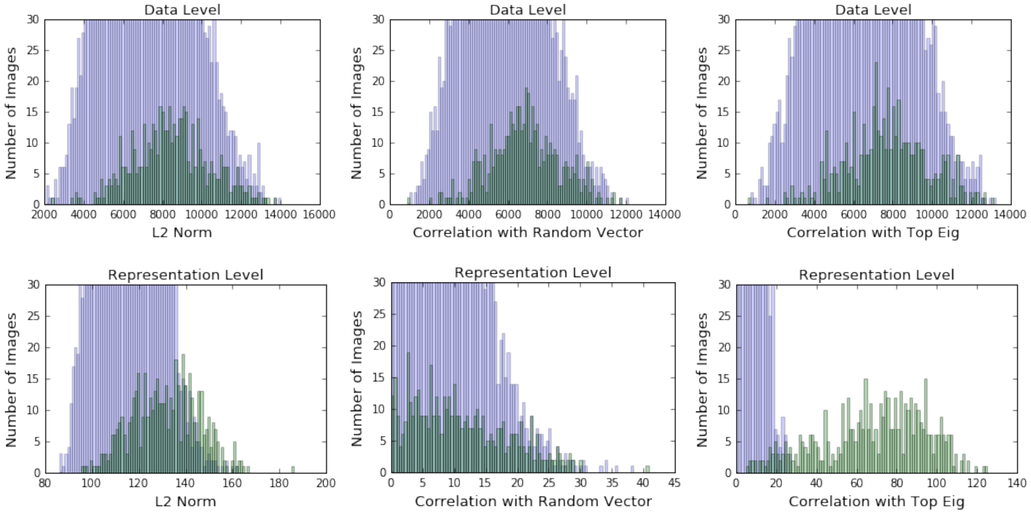


Figure 6-5: Plot of correlations for 5000 training examples correctly labelled and 500 poisoned examples incorrectly labelled. The values for the clean inputs are in blue, and those for the poisoned inputs are in green. We include plots for the computed ℓ_2 norms, correlation with a random vector, and correlation with the top singular vector of the covariance matrix of examples (respectively, representations).

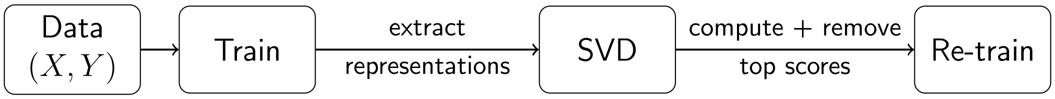


Figure 6-6: Illustration of the pipeline. We first train a neural network on the data. Then, for each class, we extract a learned representation for each input from that class. We next take the singular value decomposition of the covariance matix of these representations and use this to compute an outlier score for each example. Finally, we remove inputs with the top scores and re-train.

are causing a spectral signature at the learned representation level, we are removing mostly poisoned data points at every step.

6.6.4 Experiments

Setup We study watermark poisoning attacks on the CIFAR10 [KH09] dataset, using a standard ResNet [HZRS16] model with 3 groups of residual layers with filter sizes $[16, 16, 32, 64]$ and 5 residual units per layer. Unlike more complicated feature extractors such as autoencoders, the standard ResNet has no layer tuned to be a learned representation for any desired task. However, one can think of any of the layers as modeling different kinds of representations. For example, the first convolutional layer is typically believed to represent edges in the image while the latter layers learn "high level" features [D$^+$14]. In particular, it is common to treat the last few layers as representations for classification.

Our experiments showed that our outlier removal method successfully removes the watermark when applied on many of the later layers. We choose to report the results for the second to last residual unit simply because, on average, the method applied to this layer removed the most poisoned images. We also remark that we tried our method directly on the input. Even when data augmentation is removed, so that the watermark is not flipped or translated, the signal is still not strong enough to be detected, suggesting that a learned representation amplifying the signal is really necessary.

Attacks Our standard attack setup consists of a pair of (attack, target) labels, a watermark shape (pixel, X, or L), an epsilon (number of poisoned images), a position in the image, and a color for the mark.

For our experiments, we choose 4 pairs of labels by hand- (airplane, bird), (automobile, cat), (cat, dog), (horse, deer)- and 4 pairs randomly- (automobile, dog), (ship, frog), (truck, bird), (cat,horse). Then, for each pair of labels, we generate a random shape, position, and color for the watermark. We also use the hand-chosen watermarks of Figure 6-4.

Attack Statistics Here, we show some statistics from the attacks that give motivation for why our method works. First, in the bottom right plot of Figure 6-5, we can see a clear separation between the scores of the poisoned images and those of the clean images. This is reflected in the statistics displayed in Table 6.1. Here, we record the norms of the mean of the representation vectors for both the clean inputs as well as the clean plus watermarked inputs. Then, we record the norm of the difference in mean to measure the shift created by adding the poisoned examples. Similarly, we have the top three singular values for the mean-shifted matrix of representation vectors of both the clean examples and the clean plus watermarked examples. We can see from the table that there is quite a significant increase in the singular values upon addition of the poisoned examples. The statistics gathered suggest that our outlier detection algorithm should succeed in removing the poisoned inputs.

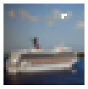
Table 6.1: We record statistics for the two experiments coming from Figure 6-4, watermarked planes labelled as birds and watermarked cars labelled as cats. For both the clean dataset and the clean plus poisoned dataset, we record the norm of the mean of the representation vectors and the top three singular values of the covariance matrix formed by these vectors. We also record the norm of the difference in the means of the vectors from the two datasets.

Experiment	Norm of Mean	Shift in Mean	1st SV	2nd SV	3rd SV
Birds only	78.751	N/A	1194.223	1115.931	967.933
Birds + planes	78.855	6.194	1613.486	1206.853	1129.711
Cats + cars	89.409	N/A	1016.919	891.619	877.743
Cats + poison	89.690	7.343	1883.934	1030.638	913.895

Evaluating our Method In Table 6.2, we record the results for a selection of our training iterations. For each experiment, we record the accuracy on the natural evaluation set (all 10000 test images for CIFAR10) as well as the poisoned evaluation set (1000 images of the attack label with a watermark). We then record the number of poisoned images left after one removal step and the accuracies upon retraining. The table shows that for a variety of parameter choices, the method successfully removes the attack. Specifically, the clean and poisoned test accuracies for the second training iteration after the removal step are comparable to those achieved by a standard trained network on a clean dataset. For reference, a standard trained network on a clean training set classifies a clean test set with accuracy 92.67% and classifies each poisoned test set with accuracy given in the rightmost column of Table 6.2. We refer the reader to Figure F.1 in the appendix for results from more choices of attack parameters.

We also reran the experiments multiple times with different random choices for the attacks. For each run that successfully captured the watermark in the first iteration, which we define as recording approximately 90% or higher accuracy on the poisoned set, the results were similar to those recorded in the table. As an aside, we note that 5% poisoned images is not enough to capture the watermark according to our definition in our examples from Figure 6-4, but 10% is sufficient.

Table 6.2: Main results for a selection of different attack parameters. Natural and poisoned accuracy are reported for two iterations, before and after the removal step. We compare to the accuracy on each poisoned test set obtained from a network trained on a clean dataset (Std Pois). The attack parameters are given by a watermarked attack image, target label, and percentage of added images.

Sample	Target	Epsilon	Nat 1	Pois 1	# Pois Left	Nat 2	Pois 2	Std Pois
	bird	5%	92.27%	74.20%	57	92.64%	2.00%	1.20%
		10%	92.32%	89.80%	7	92.68%	1.50%	
	cat	5%	92.45%	83.30%	24	92.24%	0.20%	0.10%
		10%	92.39%	92.00%	0	92.44%	0.00%	
	dog	5%	92.17%	89.80%	7	93.01%	0.00%	0.00%
		10%	92.55%	94.30%	1	92.64%	0.00%	
	horse	5%	92.60%	99.80%	0	92.57%	1.00%	0.80%
		10%	92.26%	99.80%	0	92.63%	1.20%	
	cat	5%	92.86%	98.60%	0	92.79%	8.30%	8.00%
		10%	92.29%	99.10%	0	92.57%	8.20%	
	deer	5%	92.68%	99.30%	0	92.68%	1.10%	1.00%
		10%	92.68%	99.90%	0	92.74%	1.60%	
	frog	5%	92.87%	88.80%	10	92.61%	0.10%	0.30%
		10%	92.82%	93.70%	3	92.74%	0.10%	
	bird	5%	92.52%	97.90%	0	92.69%	0.00%	0.00%
		10%	92.68%	99.30%	0	92.45%	0.50%	

Chapter 7

Filtering III: Robust Stochastic Optimization

As I watch the cherry blossoms fading, falling one by one,
I worry that your feelings will slowly die too.
Quietly, this gentle spring that we once shared is passing.
I close my eyes, and wonder if it must be so.

7.1 Introduction

In the previous chapter, we demonstrated the effectiveness of our algorithms for robust estimation in a variety of settings. While we have hopefully demonstrated that these algorithms can be used for many important tasks, ultimately the application of these methods is limited by the fact that they are designed for unsupervised estimation tasks. In particular, it is unclear how to use these algorithms to address questions such as robust supervised learning. More generally, the following question remains:

Is there a framework for "robustifying" general machine learning tasks?

That is, given an algorithm for some inference problem, is it possible to give another algorithm for this problem that solves this inference problem, when a small number

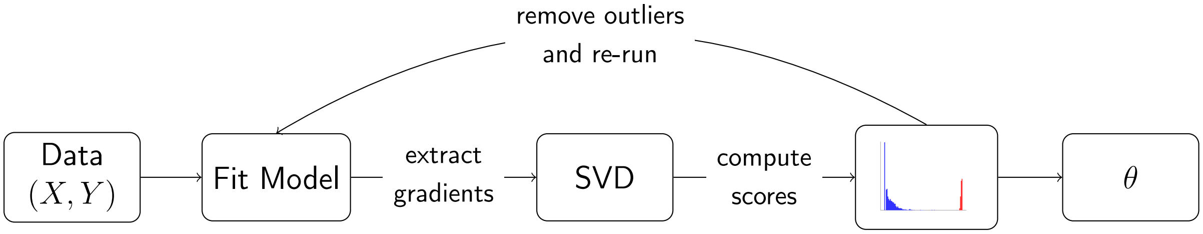


Figure 7-1: Illustration of the Sever pipeline. We first use any machine learning algorithm to fit a model to the data. Then, we extract gradients for each data point at the learned parameters, and take the singular value decomposition of the gradients. We use this to compute an outlier score for each data point. If we detect outliers, we remove them and re-run the learning algorithm; otherwise, we output the learned parameters.

of samples are corrupted? Ideally, to make it easy to apply in as many settings as possible, we should be able to do this in a black-box fashion.

In this chapter, we make progress towards answering the above question. We propose an algorithm, Sever, that is:

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

At a high level, our algorithm (depicted in Figure 7-1 and described in detail in Section 7.2.2) is a simple "plug-in" outlier detector—first, run whatever learning procedure would be run normally (e.g., least squares in the case of linear regression). Then, consider the matrix of gradients at the optimal parameters, and compute the top singular vector of this matrix. Finally, remove any points whose projection onto this singular vector is too large (and re-train if necessary).

Despite its simplicity, our algorithm possesses strong theoretical guarantees: As long as the data is not too heavy-tailed, Sever is provably robust to outliers—see Section 7.2 for detailed statements of the theory. At the same time, we show that our algorithm works very well in practice and outperforms a number of natural baseline outlier detectors. We implement our method on two tasks—a linear regression task for predicting protein activity levels [OSB^{+}18], and a spam classification task based on e-mails from the Enron corporation [MAP06]. Even with a small fraction of outliers, baseline methods perform extremely poorly on these datasets; for instance, on the Enron spam dataset with a 1% fraction of outliers, baseline errors range from 13.4% to 20.5%, while Sever incurs only 7.3% error (in comparison, the error is 3% in the absence of outliers). Similarly, on the drug design dataset, with 10% corruptions, we achieved 1.42 mean-squared error test error, compared to 1.51-2.33 for the baselines, and 1.23 error on the uncorrupted dataset.

7.2 Framework and algorithm

In this section, we describe our formal framework as well as the Sever algorithm.

7.2.1 Formal setting

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

To help us learn the parameter vector w^*, we have access to a *training set* of n functions $f_{1:n} =: \{f_1, \ldots, f_n\}$. (For linear regression, we would have $f_i(w) = \frac{1}{2}(w \cdot x_i - y_i)^2$, where (x_i, y_i) is an observed data point.) However, unlike the classical (uncorrupted) setting where we assume that $f_1, \ldots, f_n \sim p^*$, we will assume that these samples are ε-corrupted from p^*.

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

Definition 7.2.1. Given a function $f : \mathcal{H} \to \mathbb{R}$, a γ-approximate critical point of f, is a point $w \in \mathcal{H}$ so that for all unit vectors v where $w + \delta v \in \mathcal{H}$ for arbitrarily small positive δ, we have that $v \cdot \nabla f(w) \geq -\gamma$.

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

We are now ready to define the notion of a γ-*approximate* learner:

Definition 7.2.2. A learning algorithm $\mathcal{L}$ is called γ-*approximate* if, for any functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$ each bounded below on a closed domain $\mathcal{H}$, the output $w = \mathcal{L}(f_{1:n})$ of $\mathcal{L}$ is a γ-approximate critical point of $f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)$.

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

7.2.2 Algorithm and theory

As outlined in Figure 6-6, our algorithm works by post-processing the gradients of a black-box learning algorithm. The basic intuition is as follows: we want to ensure that the outliers do not have a large effect on the learned parameters. Intuitively, for the outliers to have such an effect, their corresponding gradients should be (i) large in magnitude and (ii) systematically pointing in a specific direction. We can detect this via singular value decomposition–if both (i) and (ii) hold then the outliers should be responsible for a large singular value in the matrix of gradients, which allows us to detect and remove them.

This is shown more formally via the pseudocode in Algorithm 31.

Algorithm 31 SEVER($f_{1:n}, \mathcal{L}, \sigma$)

```
Initialize $S \leftarrow \{1, \ldots, n\}$.
repeat
    $w \leftarrow \mathcal{L}(\{f_i\}_{i \in S})$. ▷ Run approximate learner on points in $S$.
    Let $\widehat{\nabla} = \frac{1}{|S|} \sum_{i \in S} \nabla f_i(w)$.
    Let $G = [\nabla f_i(w) - \widehat{\nabla}]_{i \in S}$ be the $|S| \times d$ matrix of centered gradients.
    Let $v$ be the top right singular vector of $G$.
    Compute the vector $\tau$ of outlier scores defined via $\tau_i = \left((\nabla f_i(w) - \widehat{\nabla}) \cdot v\right)^2$.
    $S' \leftarrow S$
    $S \leftarrow$ SPECTRALFILTER($S', \tau, \sigma$, False, SECONDMOMENTREMOVE) ▷ Remove
  some $i$'s with the largest scores $\tau_i$ from $S$ using the second moment filter.
until $S = S'$.
Return $w$.
```

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

Theorem 7.2.1. *Suppose that functions $f_1, \ldots, f_n, \overline{f} : \mathcal{H} \to \mathbb{R}$ are bounded below on a closed domain $\mathcal{H}$, and suppose that they satisfy the following deterministic regularity conditions: There exists a set $S_{\text{good}} \subseteq [n]$ with $|S_{\text{good}}| \geq (1 - \varepsilon)n$ and $\sigma > 0$ such that*

(i) $\text{Cov}_{S_{\text{good}}}[\nabla f_i(w)] \preceq \sigma^2 I$, *$w \in \mathcal{H}$,*

(ii) $\|\nabla \hat{f}(w) - \nabla \overline{f}(w)\|_2 \leq \sigma\sqrt{\varepsilon}$, *$w \in \mathcal{H}$, where* $\hat{f} =: (1/|S_{\text{good}}|) \sum_{i \in S_{\text{good}}} f_i$.

Then our algorithm SEVER *applied to $f_1, \ldots, f_n, \sigma$ returns a point $w \in \mathcal{H}$ that, with probability at least* $9/10$*, is a $(\gamma + O(\sigma\sqrt{\varepsilon}))$-approximate critical point of $\overline{f}$.*

The key take-away from Theorem 7.2.1 is that the error guarantee has no dependence on the underlying dimension d. In contrast, most natural algorithms incur an error that grows with d, and hence have poor robustness in high dimensions.

We show that under some niceness assumptions on p^*, the deterministic regularity conditions are satisfied with high probability with polynomially many samples:

Proposition 7.2.2 (Informal). *Let $\mathcal{H} \subset \mathbb{R}^d$ be a closed bounded set with diameter at most r. Let p^* be a distribution over functions $f : \mathcal{H} \to \mathbb{R}$ and $\overline{f} =$*

$\mathbb{E}_{f\sim p^*}[f]$. *Suppose that for each $w \in \mathcal{H}$ and unit vector v we have $\mathbb{E}_{f\sim p^*}[(v\cdot(\nabla f(w) - \overline{f}(w)))^2] \le \sigma^2$. Under appropriate Lipschitz and smoothness assumptions, for $n = \Omega(d\log(r/(\sigma^2\varepsilon))/(\sigma^2\varepsilon))$, an ε-corrupted set of functions drawn i.i.d. from p^*, $f_1,\ldots,f_n$ with high probability satisfy conditions (i) and (ii).*

The reader is referred to Proposition 7.3.5 for a detailed formal statement.

While Theorem 7.2.1 is very general and holds even for non-convex loss functions, we might in general hope for more than an approximate critical point. In particular, for convex problems, we can guarantee that we find an approximate global minimum. This follows as a corollary of Theorem 7.2.1:

Corollary 7.2.3. *Suppose that $f_1,\ldots,f_n : \mathcal{H}\to\mathbb{R}$ satisfy the regularity conditions (i) and (ii), and that $\mathcal{H}$ is convex with ℓ_2-radius r. Then, with probability at least* $9/10$, *the output of* Sever *satisfies the following:*

(i) If $\overline{f}$ is convex, the algorithm finds a $w\in\mathcal{H}$ such that $\overline{f}(w)-\overline{f}(w^*) = O((\sigma\sqrt{\varepsilon}+\gamma)r)$.

(ii) If $\overline{f}$ is ξ-strongly convex, the algorithm finds a $w\in\mathcal{H}$ such that $\overline{f}(w)-\overline{f}(w^*) = O\left((\varepsilon\sigma^2+\gamma^2)/\xi\right)$.

Practical Considerations. For our theory to hold, we need to use the randomized filtering algorithm described in Section 5.5, and filter until the stopping condition in line 10 of Algorithm 31 is satisfied. However, in practice we found that the following simpler algorithm worked well: in each iteration simply remove the top p fraction of outliers according to the scores τ_i, and instead of using a specific stopping condition, simply repeat the filter for r iterations in total. This is the version of Sever that we use in our experiments in Section 7.5.

7.2.3 Overview of Sever and its analysis

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

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

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

In summary, the first algorithmic idea is to use a robust mean estimation routine as a black-box in order to robustly estimate the gradient at *each* iteration of (projected) gradient descent. This yields a simple robust method for stochastic optimization with polynomial sample complexity and running time in a very general setting. However, this is somewhat cumbersome to run in practice. Indeed, because a single iteration of this robust gradient descent method would require a full pass over the data, in most modern settings the runtime would be prohibitively high. This is described in more detail in [DKK$^+$18b], but for the sake of conciseness we omit this description here.

We are now ready to describe SEVER (Algorithm 31) and the main insight behind it. Roughly speaking, SEVER only calls our robust mean estimation routine (which is essentially the filtering method of [DKK+17] for outlier removal) each time the algorithm reaches an approximate critical point of $\hat{f}$. There are two main motivations for this approach: First, we empirically observed that if we iteratively filter samples, keeping the subset with the samples removed, then few iterations of the filter remove points. Second, an iteration of the filter subroutine is more expensive than an iteration of gradient descent. Therefore, it is advantageous to run many steps of gradient descent on the current set of corrupted samples between consecutive filtering steps. This idea is further improved by using stochastic gradient descent, rather than computing the average at each step.

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

7.3 General analysis of SEVER

This section is dedicated to the analysis of Algorithm 31, where we do not make convexity assumptions about the underlying functions $f_1, \ldots, f_n$. In this case, we can show that our algorithm finds an approximate critical point of $\overline{f}$. When we specialize to convex functions, this immediately implies that we find an approximate minimal point of $\overline{f}$.

Our proof proceeds in two parts. First, we define a set of deterministic conditions under which our algorithm finds an approximate minimal point of $\overline{f}$. We then show that, under mild assumptions on our functions, this set of deterministic conditions holds with high probability after polynomially many samples.

Deterministic conditions We first explicitly demonstrate a set of deterministic conditions on the (uncorrupted) data points. Our deterministic regularity conditions are as follows:

Assumption 7.3.1. Fix $0 < \varepsilon < 1/2$. There exists an unknown set $S_{\mathrm{good}} \subseteq [n]$ with $|S_{\mathrm{good}}| \geq (1-\varepsilon)n$ of "good" functions $\{f_i\}_{i \in S_{\mathrm{good}}}$ and parameters $\sigma_0, \sigma_1 \in \mathbb{R}_+$ such that:

$$\left\| \mathop{\mathbb{E}}_{S_{\mathrm{good}}} \left[\left(\nabla f_i(w) - \nabla \overline{f}(w)\right)\left(\nabla f_i(w) - \nabla \overline{f}(w)\right)^T \right] \right\|_2 \leq (\sigma_0 + \sigma_1 \|w^* - w\|_2)^2, \text{ for all } w \in \mathcal{H}\ , \tag{7.1}$$

and

$$\|\nabla \hat{f}(w) - \nabla \overline{f}(w)\|_2 \leq (\sigma_0 + \sigma_1 \|w^* - w\|_2)\sqrt{\varepsilon}, \text{ for all } w \in \mathcal{H}, \text{ where } \hat{f} =: \frac{1}{|S_{\mathrm{good}}|} \sum_{i \in S_{\mathrm{good}}} f_i\ . \tag{7.2}$$

In Section 7.3.1, we prove the following theorem, which shows that under Assumption 7.3.1 our algorithm succeeds:

Theorem 7.3.2. *Suppose that the functions $f_1, \ldots, f_n, \overline{f} : \mathcal{H} \to \mathbb{R}$ are bounded below, and that Assumption 7.3.1 is satisfied, where $\sigma =: \sigma_0 + \sigma_1\|w^* - w\|_2$. Then* Sever *applied to $f_1, \ldots, f_n, \sigma$ returns a point $w \in \mathcal{H}$ that, with probability at least* 9/10, *is a $(\gamma + O(\sigma\sqrt{\varepsilon}))$-approximate critical point of $\overline{f}$.*

Observe that the above theorem holds quite generally; in particular, it holds for non-convex functions. As a corollary of this theorem, in Section 7.3.2 we show that this immediately implies that Sever robustly minimizes convex functions, if Assumption 7.3.1 holds:

Corollary 7.3.3. *For functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$, suppose that Assumption 7.3.1 holds and that $\mathcal{H}$ is convex. Then, with probability at least* 9/10, *for some universal* *constant ε_0, if $\varepsilon < \varepsilon_0$, the output of* Sever *satisfies the following:*

(i) If $\overline{f}$ is convex, the algorithm finds a $w \in \mathcal{H}$ such that $\overline{f}(w) - \overline{f}(w^*) = O((\sigma_0 r + \sigma_1 r^2)\sqrt{\varepsilon} + \gamma r)$.

(ii) If $\overline{f}$ is ξ-strongly convex, the algorithm finds a $w \in \mathcal{H}$ such that

$$\overline{f}(w) - \overline{f}(w^*) = O\left(\frac{\varepsilon}{\xi}(\sigma_0 + \sigma_1 r)^2 + \frac{\gamma^2}{\xi}\right) .$$

In the strongly convex case and when $\sigma_1 > 0$, we can remove the dependence on σ_1 and r in the above by repeatedly applying Sever with decreasing r:

Corollary 7.3.4. *For functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$, suppose that Assumption 7.3.1 holds, that $\mathcal{H}$ is convex and that $\overline{f}$ is ξ-strongly convex for $\xi \geq C\sigma_1\sqrt{\varepsilon}$ for some absolute constant C. Then, with probability at least $9/10$, for some universal constant ε_0, if $\varepsilon < \varepsilon_0$, we can find a $\widehat{w}$ with*

$$\overline{f}(\widehat{w}) - \overline{f}(w^*) = O\left(\frac{\varepsilon\sigma_0^2 + \gamma^2}{\xi}\right) .$$

and

$$\|\widehat{w} - w^*\|_2 = O\left(\frac{\sqrt{\varepsilon}\sigma_0 + \gamma}{\xi}\right)$$

using at most $O(\log(r\xi/(\gamma + \sigma_0\sqrt{\varepsilon})))$ calls to Sever.

To concretely use Theorem 7.3.2, Corollary 7.3.3, and Corollary 7.3.4, in Section 7.3.4 we show that the Assumption 7.3.1 is satisfied with high probability under mild conditions on the distribution over the functions, after drawing polynomially many samples:

Proposition 7.3.5. *Let $\mathcal{H} \subset \mathbb{R}^d$ be a closed bounded set with diameter at most r. Let p^* be a distribution over functions $f : \mathcal{H} \to \mathbb{R}$ with $\overline{f} = \mathbb{E}_{f\sim p^*}[f]$ so that $f - \overline{f}$ is L-Lipschitz and β-smooth almost surely. Assume furthermore that for each $w \in \mathcal{H}$*

and unit vector v that $\mathbb{E}_{f\sim p^*}[(v \cdot (\nabla f(w) - \overline{f}(w)))^2] \leq \sigma^2/2$. Then for

$$n = \Omega\left(\frac{dL^2 \log(r\beta L/\sigma^2\varepsilon)}{\sigma^2\varepsilon}\right) ,$$

an ε-corrupted set of points $f_1, \ldots, f_n$ with high probability satisfy Assumption 7.3.1.

The remaining subsections are dedicated to the proofs of Theorem 7.3.2, Corollary 7.3.3, Corollary 7.3.4, and Proposition 7.3.5.

7.3.1 Proof of Theorem 7.3.2

Throughout this proof we let S_{good} be as in Assumption 7.3.1. We require the following two lemmata. Roughly speaking, the first states that on average, we remove more corrupted points than uncorrupted points, and the second states that at termination, and if we have not removed too many points, then we have reached a point at which the empirical gradient is close to the true gradient. Formally:

Lemma 7.3.6. *If the samples satisfy* (7.1) *of Assumption 7.3.1, and if $|S| \geq 2n/3$ then if S' is the output of Line 9, we have that*

$$\mathbb{E}[|S_{\text{good}} \cap (S \backslash S')|] \leq \mathbb{E}[|([n] \backslash S_{\text{good}}) \cap (S \backslash S')|].$$

Lemma 7.3.7. *If the samples satisfy Assumption 7.3.1,* $\textsc{Filter}(S, \tau, \sigma) = S$, *and $n - |S| \leq 11\varepsilon n$, then*

$$\left\| \nabla \overline{f}(w) - \frac{1}{|S_{\text{good}}|} \sum_{i \in S} \nabla f_i(w) \right\|_2 \leq O(\sigma\sqrt{\varepsilon})$$

Before we prove these lemmata, we show how together they imply Theorem 7.3.2.

Proof of Theorem 7.3.2 assuming Lemma 7.3.6 and Lemma 7.3.7. First, we note that the algorithm must terminate in at most n iterations. This is easy to see as each iteration of the main loop except for the last must decrease the size of S by at least 1.

It thus suffices to prove correctness. Note that Lemma 7.3.6 says that each iteration will on average throw out as many elements not in S_{good} from S as elements in S_{good}. In particular, this means that $|([n]\backslash S_{\text{good}}) \cap S| + |S_{\text{good}} \backslash S|$ is a supermartingale. Since its initial size is at most εn, with probability at least 9/10, it never exceeds $10\varepsilon n$, and therefore at the end of the algorithm, we must have that $n - |S| \leq \varepsilon n + |S_{\text{good}} \backslash S| \leq 11\varepsilon n$. This will allow us to apply Lemma 7.3.7 to complete the proof, using the fact that w is a γ-approximate critical point of $\frac{1}{|S_{\text{good}}|} \sum_{i \in S} \nabla f_i(w)$. □

Thus it suffices to prove these two lemmata. We first prove Lemma 7.3.6:

Proof of Lemma 7.3.6. Let $S_{\text{good}} = S \cap S_{\text{good}}$ and $S_{\text{bad}} = S \backslash S_{\text{good}}$. We wish to show that the expected number of elements thrown out of S_{bad} is at least the expected number thrown out of S_{good}. We note that our result holds trivially if $\textsc{Filter}(S, \tau, \sigma) = S$. Thus, we can assume that $\mathbb{E}_{i \in S}[\tau_i] \geq 12\sigma$.

It is easy to see that the expected number of elements thrown out of S_{bad} is proportional to $\sum_{i \in S_{\text{bad}}} \tau_i$, while the number removed from S_{good} is proportional to $\sum_{i \in S_{\text{good}}} \tau_i$ (with the same proportionality). Hence, it suffices to show that $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$.

We first note that since $\text{Cov}_{i \in S_{\text{good}}}[\nabla f_i(w)] \preceq \sigma^2 I$, we have that

$$\begin{aligned} \operatorname*{Cov}_{i \in S_{\text{good}}}[v \cdot \nabla f_i(w)] &\overset{(a)}{\leq} \frac{3}{2} \operatorname*{Cov}_{i \in S_{\text{good}}}[v \cdot \nabla f_i(w)] \\ &= \frac{3}{2} \cdot v^\top \operatorname*{Cov}_{i \in S_{\text{good}}}[\nabla f_i(w)] v \leq 2\sigma^2 , \end{aligned}$$

where (a) follows since $|S_{\text{good}}| \geq \frac{3}{2} S_{\text{good}}$.

Let $\mu_{\text{good}} = \mathbb{E}_{i \in S_{\text{good}}}[v \cdot \nabla f_i(w)]$ and $\mu = \mathbb{E}_{i \in S}[v \cdot \nabla f_i(w)]$. Note that

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

We now split into two cases.

Firstly, if $(\mu - \mu_{\text{good}})^2 \geq 4\sigma^2$, we let $\mu_{\text{bad}} = \mathbb{E}_{i \in S_{\text{bad}}}[v \cdot \nabla f_i(w)]$, and note that

$|\mu - \mu_{\text{bad}}||S_{\text{bad}}| = |\mu - \mu_{\text{good}}||S_{\text{good}}|$. We then have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{i \in S_{\text{bad}}}[\tau_i] &\geq (\mu - \mu_{\text{bad}})^2 \\
&\geq (\mu - \mu_{\text{good}})^2 \left(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|} \right)^2 \\
&\geq 2 \left(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|} \right) (\mu - \mu_{\text{good}})^2 \\
&\geq \left(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|} \right) \mathop{\mathbb{E}}_{i \in S_{\text{good}}}[\tau_i].
\end{aligned}$$

Hence, $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$.

On the other hand, if $(\mu - \mu_{\text{good}})^2 \leq 4\sigma^2$, then $\mathbb{E}_{i \in S_{\text{good}}}[\tau_i] \leq 6\sigma^2 \leq \mathbb{E}_{i \in S}[\tau_i]/2$. Therefore $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$ once again. This completes our proof. □

We now prove Lemma 7.3.7.

Proof of Lemma 7.3.7. We need to show that

$$\delta := \left\| \sum_{i \in S} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 = O(n\sigma\sqrt{\varepsilon}).$$

We note that

$$\begin{aligned}
&\left\| \sum_{i \in S} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 \\
&\leq \left\| \sum_{i \in S_{\text{good}}} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 + \left\| \sum_{i \in (S_{\text{good}} \setminus S)} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 + \left\| \sum_{i \in (S \setminus S_{\text{good}})} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 \\
&= \left\| \sum_{i \in (S_{\text{good}} \setminus S)} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 + \left\| \sum_{i \in (S \setminus S_{\text{good}})} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 + O(n\sqrt{\sigma^2\varepsilon}).
\end{aligned}$$

First we analyze

$$\left\| \sum_{i \in (S_{\text{good}} \setminus S)} (\nabla f_i(w) - \nabla \overline{f}(w)) \right\|_2 .$$

This is the supremum over unit vectors v of

$$\sum_{i \in (S_{\text{good}} \backslash S)} v \cdot (\nabla f_i(w) - \nabla \overline{f}(w)).$$

However, we note that

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

Since $|S_{\text{good}} \backslash S| = O(n\varepsilon)$, we have by Cauchy-Schwarz that

$$\sum_{i \in (S_{\text{good}} \backslash S)} v \cdot (\nabla f_i(w) - \nabla \overline{f}(w)) = O(\sqrt{(n\sigma^2)(n\varepsilon)}) = O(n\sqrt{\sigma^2\varepsilon}),$$

as desired.

We note that since for any such v that

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

(or otherwise our filter would have removed elements) and since $|S \backslash S_{\text{good}}| = O(n\varepsilon)$, and so we have similarly that

$$\left\| \sum_{i \in (S \backslash S_{\text{good}})} \nabla f_i(w) - \nabla \overline{f}(w) \right\|_2 = O(n\sigma\sqrt{\varepsilon} + \delta\sqrt{n\varepsilon}).$$

Combining with the above we have that

$$\delta = O(\sigma\sqrt{\varepsilon} + \delta\sqrt{\varepsilon/n}),$$

and therefore, $\delta = O(\sigma\sqrt{\varepsilon})$ as desired. □

7.3.2 Proof of Corollary 7.3.3

In this section, we show that the Sever algorithm finds an approximate global optimum for convex optimization in various settings, under Assumption 7.3.1. We do so by simply applying the guarantees of Theorem 7.3.2 in a fairly black box manner.

Before we proceed with the proof of Corollary 7.3.3, we record a simple lemma that allows us to translate an approximate critical point guarantee to an approximate global optimum guarantee:

Lemma 7.3.8. *Let $f : \mathcal{H} \to \mathbb{R}$ be a convex function and let $x \neq y \in \mathcal{H}$. Let $v = y - x/\|y-x\|_2$ be the unit vector in the direction of $y - x$. Suppose that for some δ that $v \cdot (\nabla f(x)) \geq -\delta$ and $-v \cdot (\nabla f(y)) \geq -\delta$. Then we have that:*

1. *$|f(x) - f(y)| \leq \|x - y\|_2 \delta$.*
2. *If f is ξ-strongly convex, then $|f(x) - f(y)| \leq 2\delta^2/\xi$ and $\|x - y\|_2 \leq 2\delta/\xi$.*

Proof. Let $r = \|x - y\|_2 > 0$ and $g(t) = f(x + tv)$. We have that $g(0) = f(x), g(r) = f(y)$ and that g is convex (or ξ-strongly convex) with $g'(0) \geq -\delta$ and $g'(r) \leq \delta$. By convexity, the derivative of g is increasing on $[0, r]$ and therefore $|g'(t)| \leq \delta$ for all $t \in [0, r]$. This implies that

$$|f(x) - f(y)| = |g(r) - g(0)| = \left| \int_0^r g'(t) dt \right| \leq r\delta \; .$$

To show the second part of the lemma, we note that if g is ξ-strongly convex that $g''(t) \geq \xi$ for all t. This implies that $g'(r) > g'(0) + \xi r$. Since $g'(r) - g'(0) \leq 2\delta$, we obtain that $r \leq 2\delta/\xi$, from which the second statement follows. □

Proof of Corollary 7.3.3. By applying the algorithm of Theorem 7.3.2, we can find a point w that is a $\gamma' =: (\gamma + O(\sigma\sqrt{\varepsilon}))$-approximate critical point of $\overline{f}$, where $\sigma =: \sigma_0 + \sigma_1 \|w^* - w\|_2$. That is, for any unit vector v pointing towards the interior of $\mathcal{H}$, we have that $v \cdot \nabla \overline{f}(w) \geq -\gamma'$.

To prove (i), we apply Lemma 7.3.8 to $\overline{f}$ at w which gives that

$$|\overline{f}(w) - \overline{f}(w^*)| \leq r \cdot \gamma'.$$

To prove (ii), we apply Lemma 7.3.8 to $\overline{f}$ at w which gives that

$$|\overline{f}(w) - \overline{f}(w^*)| \leq 2\gamma'^2/\xi.$$

Plugging in parameters appropriately then immediately gives the desired bound. □

7.3.3 Proof of Corollary 7.3.4

We apply SEVER iteratively starting with a domain $\mathcal{H}_1 = \mathcal{H}$ and radius $r_1 = r$. After each iteration, we know the resulting point is close to w^* will be able to reduce the search radius.

At step i, we have a domain of radius r_i. As in the proof of Corollary 7.3.3 above, we apply algorithm of Theorem 7.3.2, we can find a point w_i that is a $\gamma_i' =: (\gamma + O(\sigma_i'\sqrt{\varepsilon}))$-approximate critical point of $\overline{f}$, where $\sigma_i' =: \sigma_0 + \sigma_1 r_i$. Then using Lemma 7.3.8, we obtain that $\|w_i - w^*\|_2 \leq 2\gamma_i'/\xi$.

Now we can define $\mathcal{H}_{i+1}$ as the intersection of $\mathcal{H}$ and the ball of radius $r_{i+1} = 2\gamma_i'/\xi$ around w_i and repeat using this domain. We have that $r_{i+1} = 2\gamma_i'/\xi = 2\gamma/\xi + O(\sigma_0\sqrt{\varepsilon}/\xi + \sigma_1\sqrt{\varepsilon}r_i/\xi)$. Now if we choose the constant C such that the constant in this $O()$ is $C/4$, then using our assumption that $\xi \geq 2\sigma_1\sqrt{\varepsilon}$, we obtain that

$$r_{i+1} \leq 2\gamma/\xi + C\sigma_0\sqrt{\varepsilon}/4\xi + C\sigma_1\sqrt{\varepsilon}r_i/4\xi \leq 2\gamma/\xi + C\sigma_0\sqrt{\varepsilon}/4 + r_i/4$$

Now if $r_i \geq 8\gamma/\xi + 2C\sigma_0\sqrt{\varepsilon}/\xi$, then we have $r_{i+1} \leq r_i/2$ and if $r_i \leq 8\gamma/\xi + 2C\sigma_0\sqrt{\varepsilon}/\xi$ then we also have $r_{i+1} \leq 8\gamma/\xi + 2C\sigma_0\sqrt{\varepsilon}/\xi$. When r_i is smaller than this we stop and output w_i. Thus we stop in at most $O(\log(r) - \log(8\gamma/\xi + 2C\sigma_0\sqrt{\varepsilon}/\xi)) = O(\log(r\xi/(\gamma + \sigma_0\sqrt{\varepsilon}))$ iterations and have $r_i = O(\gamma/\xi + C\sigma_0\sqrt{\varepsilon})$. But then $\gamma_i' = \gamma + O(\sigma_i'\sqrt{\varepsilon})) \leq \gamma + C(\sigma_0 + \sigma_1 r_i')\sqrt{\varepsilon}/8 = O(\gamma + \sigma_0\sqrt{\varepsilon})$. Using Lemma 7.3.8 we obtain that

$$|\overline{f}(w_i) - \overline{f}(w^*)| \leq 2\gamma_i'^2/\xi = O(\gamma^2/\xi + \sigma_0^2\varepsilon/\xi).$$

as required. The bound on $\|\widehat{w} - w^*\|_2$ follows similarly.

Remark 7.3.1. While we don't give explicit bounds on the number of calls to the approximate learner needed by SEVER, such bounds can be straightforwardly obtained under appropriate assumptions on the f_i (see, e.g., the following subsection). Two remarks are in order. First, in this case we cannot take advantage of assumptions that only hold at $\overline{f}$ but might not on the corrupted average f. Second, our algorithm can take advantage of a closed form for the minimum. For example, for the case of linear regression, f_i is not Lipschitz with a small constant if x_i is far from the mean, but there is a simple closed form for the minimum of the least squares loss.

7.3.4 Proof of Proposition 7.3.5

We let S_{good} be the set of uncorrupted functions f_i. It is then the case that $|S_{\text{good}}| \geq (1-\varepsilon)n$. We need to show that for each $w \in \mathcal{H}$ that

$$\operatorname*{Cov}_{i \in S_{\text{good}}}[\nabla f_i(w)] \leq 3\sigma^2 I/4 \tag{7.3}$$

and

$$\left\| \nabla \overline{f}(w) - \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} \nabla f_i(w) \right\|_2 \leq O(\sigma^2 \sqrt{\varepsilon}). \tag{7.4}$$

We will proceed by a cover argument. First we claim that for each $w \in \mathcal{H}$ that (7.3) and (7.4) hold with high probability. For Equation (7.3), it suffices to show that for each unit vector v in a cover $\mathcal{N}$ of size $2^{O(d)}$ of the sphere that

$$\operatorname*{\mathbb{E}}_{i \in S_{\text{good}}}[(v \cdot (\nabla f_i(w) - \overline{f}))^2] \leq 2\sigma^2/3. \tag{7.5}$$

However, we note that

$$\operatorname*{\mathbb{E}}_{p^*}[(v \cdot (\nabla f(w) - \overline{f}))^2] \leq \sigma^2/2.$$

Since $|v \cdot (\nabla f(w) - \overline{f})|$ is always bounded by L, Equation (7.5) holds for each v, w with probability at least $1 - \exp(-\Omega(n\sigma^2/L^2))$ by a Chernoff bound (noting that the removal of an ε-fraction of points cannot increase this by much). Similarly, to show

Equation 7.4, it suffices to show that for each such v that

$$\mathop{\mathbb{E}}_{i\in S_{\mathrm{good}}}[(v\cdot(\nabla f_i(w)-\overline{f}))]\leq O(\sigma\sqrt{\varepsilon}). \tag{7.6}$$

Noting that

$$\mathop{\mathbb{E}}_{p^*}[(v\cdot(\nabla f(w)-\overline{f}))]=0$$

A Chernoff bound implies that with probability $1-\exp(-\Omega(n\sigma^2\varepsilon/L^2))$ that the average over our original set of f's of $(v\cdot(\nabla f(w)-\overline{f}))$ is $O(\sigma\sqrt{\varepsilon})$. Assuming that Equation (7.5) holds, removing an ε-fraction of these f's cannot change this value by more than $O(\sigma\sqrt{\varepsilon})$. By union bounding over $\mathcal{N}$ and standard net arguments, this implies that Equations (7.3) and (7.4) hold with probability $1-\exp(\Omega(d-n\sigma^2\varepsilon/L^2))$ for any given w.

To show that our conditions hold for all $w\in\mathcal{H}$, we note that by β-smoothness, if Equation (7.4) holds for some w, it holds for all other w' in a ball of radius $\sqrt{\sigma^2\varepsilon}/\beta$ (up to a constant multiplicative loss). Similarly, if Equation (7.3) holds at some w, it holds with bound $\sigma^2 I$ for all w' in a ball of radius $\sigma^2/(2L\beta)$. Therefore, if Equations (7.3) and (7.4) hold for all w in a $\min(\sqrt{\sigma^2\varepsilon}/\beta,\sigma/(2L\beta))$-cover of $\mathcal{H}$, the assumptions of Theorem 7.3.2 will hold everywhere. Since we have such covers of size $\exp(O(d\log(r\beta L/(\sigma^2\varepsilon))))$, by a union bound, this holds with high probability if

$$n=\Omega\left(\frac{dL^2\log(r\beta L/\sigma^2\varepsilon)}{\sigma^2\varepsilon}\right),$$

as claimed.

7.4 Analysis of Sever for GLMs

A case of particular interest is that of Generalized Linear Models (GLMs):

Definition 7.4.1. Let $\mathcal{H}\subseteq\mathbb{R}^d$ and $\mathcal{Y}$ be an arbitrary set. Let D_{xy} be a distribution over $\mathcal{H}\times\mathcal{Y}$. For each $Y\in\mathcal{Y}$, let $\sigma_Y:\mathbb{R}\to\mathbb{R}$ be a convex function. The *generalized linear model* (GLM) over $\mathcal{H}\times\mathcal{Y}$ with *distribution* D_{xy} and *link functions* σ_Y is the function $\overline{f} : \mathbb{R}^d \to \mathbb{R}$ defined by $\overline{f}(w) = \mathbb{E}_{X,Y}[f_{X,Y}(w)]$, where

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

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

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

This case covers a number of interesting applications, including SVMs and logistic regression. Unfortunately, the tools developed in Section 7.3 do not seem to be able to cover this case in a simple manner. In particular, it is unclear how to demonstrate that Assumption 7.3.1 holds after taking polynomially many samples from a GLM. To rectify this, in this section, we demonstrate a different deterministic regularity condition under which we show Sever succeeds, and we show that this condition holds after polynomially many samples from a GLM. Specifically, we will show that Sever succeeds under the following deterministic condition:

Assumption 7.4.1. Fix $0 < \varepsilon < 1/2$. There exists an unknown set $S_{\text{good}} \subseteq [n]$ with $|S_{\text{good}}| \geq (1 - \varepsilon)n$ of "good" functions $\{f_i\}_{i \in S_{\text{good}}}$ and parameters $\sigma_0, \sigma_2 \in \mathbb{R}_+$ such that such that the following conditions simultanously hold:

- Equation (7.1) holds with $\sigma_1 = 0$ and the same σ_0, and

- The following equations hold:

$$\|\nabla \hat{f}(w^*) - \nabla \overline{f}(w^*)\|_2 \leq \sigma_0 \sqrt{\varepsilon} \ \text{, and} \tag{7.7}$$

$$|\hat{f}(w) - \overline{f}(w)| \leq \sigma_2 \sqrt{\varepsilon}, \ \text{for all } w \in \mathcal{H} \ , \tag{7.8}$$

  where $\hat{f} =: \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} f_i$.

In this section, we will show the following two statements. The first demonstrates that Assumption 7.4.1 implies that Sever succeeds, and the second shows that Assumption 7.4.1 holds after polynomially many samples from a GLM. Formally:

Theorem 7.4.2. *For functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$, suppose that Assumption 7.4.1 holds and that $\mathcal{H}$ is convex. Then, for some universal constant ε_0, if $\varepsilon < \varepsilon_0$, there is an algorithm which, with probability at least $9/10$, finds a $w \in \mathcal{H}$ such that*

$$\overline{f}(w) - \overline{f}(w^*) = r(\gamma + O(\sigma_0 \sqrt{\varepsilon})) + O(\sigma_2 \sqrt{\varepsilon}) \ .$$

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

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

Proposition 7.4.3. *Let $\mathcal{H} \subseteq \mathbb{R}^d$ and let $\mathcal{Y}$ be an arbitrary set. Let $f_1, \ldots, f_n$ be obtained by picking f_i i.i.d. at random from a GLM $\overline{f}$ over $\mathcal{H} \times \mathcal{Y}$ with distribution D_{xy} and link functions σ_Y, where*

$$n = \Omega\left(\frac{d \log(dr/\varepsilon)}{\varepsilon}\right) \ .$$

Suppose moreover that the following conditions all hold:

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

Then with probability at least $9/10$ over the original set of samples, there is a set of $(1 - \varepsilon)n$ of the f_i that satisfy Assumption 7.4.1 on $\mathcal{H}$ with $\sigma_0 = 2$, $\sigma_1 = 0$ and $\sigma_2 = 1 + r$.and $\sigma_2 = 1 + r$.

7.4.1 Proof of Theorem 7.4.2

As before, since Sever either terminates or throws away at least one sample, clearly it cannot run for more than n iterations. Thus the runtime bound is simple, and it suffices to show correctness.

We first prove the following lemma:

Lemma 7.4.4. *Let $f_1, \ldots, f_n$ satisfy Assumption 7.4.1. Then with probability at least 9/10,* Sever *applied to $f_1, \ldots, f_n, \sigma_0$ returns a point $w \in \mathcal{H}$ which is a $(\gamma + O(\sigma_0\sqrt{\varepsilon}))$-approximate critical point of $\hat{f}$.*

Proof. We claim that the empirical distribution over $f_1, \ldots, f_n$ satisfies Assumption 7.3.1 for the function $\hat{f}$ with σ_0 as stated and $\sigma_1 = 0$, with the S_{good} in Assumption 7.3.1 being the same as in the definition of Assumption 7.4.1. Clearly these functions satisfy (7.2) (since the LHS is zero), so it suffices to show that they satisfy (7.1) Indeed, we have that for all $w \in \mathcal{H}$,

$$\mathop{\mathbb{E}}_{S_{\text{good}}}[(\nabla f_i(w) - \nabla \hat{f}(w))(\nabla f_i(w) - \nabla \hat{f}(w))^\top] \preceq \mathop{\mathbb{E}}_{S_{\text{good}}}[(\nabla f_i(w) - \nabla \overline{f}(w))(\nabla f_i(w) - \nabla \overline{f}(w))^\top] ,$$

so they satisfy (7.1), since the RHS is bounded by Assumption 7.4.1. Thus this lemma follows from an application of Theorem 7.3.2. □

With this critical lemma in place, we can now prove Theorem 7.4.2:

Proof of Theorem 7.4.2. Condition on the event that Lemma 7.4.4 holds, and let $w \in \mathcal{H}$ be the output of Sever. By Assumption 7.4.1, we know that $\hat{f}(w^*) \geq \overline{f}(w^*) - \sigma_2\sqrt{\varepsilon}$, and moreover, w^* is a $\gamma + \sigma_0\sqrt{\varepsilon}$-approximate critical point of $\hat{f}$.

Since each link function is convex, so is $\hat{f}$. Hence, by Lemma 7.3.8, since w is a $(\gamma + O(\sigma_0\sqrt{\varepsilon}))$-approximate critical point of $\hat{f}$, we have $\hat{f}(w) - \hat{f}(w^*) \leq r(\gamma + O(\sigma_0\sqrt{\varepsilon}))$. By Assumption 7.3.1, this immediately implies that $\overline{f}(w) - \overline{f}(w^*) \leq r(\gamma + O(\sigma_0\sqrt{\varepsilon})) + O(\sigma_2\sqrt{\varepsilon})$, as claimed.

The bound for strongly convex functions follows from the exact argument, except using the statement in Lemma 7.3.8 pertaining to strongly convex functions. □

7.4.2 Proof of Proposition 7.4.3

Proof. We first note that $\nabla f_{X,Y}(w) = X\sigma'_Y(w \cdot X)$. Thus, under Assumption 7.4.1, we have for any v that

$$\mathop{\mathbb{E}}_i[(v \cdot (\nabla f_i(w) - \nabla \overline{f}(w)))^2] \ll \mathop{\mathbb{E}}_i[(v \cdot \nabla f_i(w))^2] + 1 \ll \mathop{\mathbb{E}}_i[(v \cdot X_i)^2] + 1 \; .$$

In particular, since this last expression is independent of w, we only need to check this single matrix bound.

We let our good set be the set of samples with $|X| \leq 80\sqrt{d/\varepsilon}$ that were not corrupted. By Lemma 5.5.2, we know that that with 90% probability, the non-good samples make up at most an $\varepsilon/2 + \varepsilon/160$-fraction of the original samples, and that $\mathbb{E}[XX^T]$ over the good samples is at most $2I$. This proves that the spectral bound holds everywhere. Applying it to the $\nabla f_{X,Y}(w^*)$, we find also with 90% probability that the expectation over all samples of $\nabla f_{X,Y}(w^*)$ is within $\sqrt{\varepsilon}/3$ of $\nabla \overline{f}(w^*)$. Additionally, throwing away the samples with $|\nabla f_{X,Y}(w^*) - \nabla \overline{f}(w^*)| > 80\sqrt{d/\varepsilon}$ changes this by at most $\sqrt{\varepsilon}/2$. Finally, it also implies that the variance of $\nabla f_{X,Y}(w^*)$ is at most $3/2I$, and therefore, throwing away any other ε-fraction of the samples changes it by at most an additional $\sqrt{3\varepsilon/2}$.

We only need to show that $|\mathbb{E}_{i \text{ good}}[f_i(w)] - \mathbb{E}_X[f_X(w)]| \leq \sqrt{\varepsilon}$ for all $w \in \mathcal{H}$. For this we note that since the f_X and f_i are all 1-Lipschitz, it suffices to show that $|\mathbb{E}_{i \text{ good}}[f_i(w)] - \mathbb{E}_X[f_X(w)]| \leq (1+|w|)\sqrt{\varepsilon}/2$ on an $\varepsilon/2$-cover of $\mathcal{H}$. For this it suffices to show that the bound will hold pointwise except with probability $\exp(-\Omega(d\log(r/\varepsilon)))$. We will want to bound this using pointwise concentration and union bounds, but this runs into technical problems since very large values of $X \cdot w$ can lead to large values of f, so we will need to make use of the condition above that the average of $X_i X_i^T$ over our good samples is bounded by $2I$. In particular, this implies that the contribution to the average of $f_i(w)$ over the good i coming from samples where $|X_i \cdot w| \geq 10|w|/\sqrt{\varepsilon}$ is at most $\sqrt{\varepsilon}(1 + |w|)/10$. We consider the average of $f_i(w)$ over the remaining i. Note that these values are uniform random samples from $f_X(w)$ conditioned on $|X| \leq 80\sqrt{d/\varepsilon}$ and $|X_i \cdot w| < 10|w|/\sqrt{\varepsilon}$. It will suffices to show that taking n samples from this distribution has average within $(1+|w|)\sqrt{\varepsilon}/2$ of the mean with high probability. However, since $|f_X(w)| \leq O(1+|X \cdot w|)$, we have that over this distribution $|f_X(w)|$ is always $O(1+|w|)/\sqrt{\varepsilon}$, and has variance at most $O(1+|w|)^2$. Therefore, by Bernstein's Inequality, the probability that n random samples from $f_X(w)$ (with the above conditions on X) differ from their mean by more than $(1+|w|)\sqrt{\varepsilon}/2$ is

$$\exp(-\Omega(n^2(1+|w|)^2\varepsilon/((1+|w|)^2+n(1+|w|)^2))) = \exp(-\Omega(n\varepsilon)).$$

Thus, for n at least a sufficiently large multiple of $d\log(dr/\varepsilon)/\varepsilon$, this holds for all w in our cover of $\mathcal{H}$ with high probability. This completes the proof. □

7.5 Experiments

In this section we apply Sever to regression and classification problems. As our base learners, we used ridge regression and an SVM, respectively. We implemented the latter as a quadratic program, using Gurobi [Gur16] as a backend solver and YALMIP [Löf04] as the modeling language.

In both cases, we ran the base learner and then extracted gradients for each data point at the learned parameters. We then centered the gradients and ran MATLAB's `svds` method to compute the top singular vector v, and removed the top p fraction of points i with the largest *outlier score* τ_i, computed as the squared magnitude of the projection onto v (see Algorithm 31). We repeated this for r iterations in total. For classification, we centered the gradients separately (and removed points separately) for each class, which improved performance.

We compared our method to five baseline methods. These all have the same high-level form as Sever (run the base learner then filter top p fraction of points with the largest score), but use a different definition of the score τ_i for deciding which points to filter:

- **noDefense**: no points are removed.
- **l2**: remove points where the covariate x has large ℓ_2 distance from the mean.

- **loss**: remove points with large loss (measured at the parameters output by the base learner).

- **gradient**: remove points with large gradient (in ℓ_2-norm).

- **gradientCentered**: remove points whose gradients are far from the mean gradient in ℓ_2-norm.

Note that **gradientCentered** is similar to our method, except that it removes large gradients in terms of ℓ_2-norm, rather than in terms of projection onto the top singular vector. As before, for classification we compute these metrics separately for each class.

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

For the sake of the readability of the graphs, in the figures below, we only present a small set of representative baselines. For additional plots, we refer the reader to Appendix G.

7.5.1 Ridge regression

For ridge regression, we tested our method on a synthetic Gaussian dataset as well as a drug discovery dataset. The synthetic dataset consists of observations (x_i, y_i) where $x_i \in \mathbb{R}^{500}$ has independent standard Gaussian entries, and $y_i = \langle x_i, w^* \rangle + 0.1 z_i$, where z_i is also Gaussian. We generated 5000 training points and 100 test points. The drug discovery dataset was obtained from the ChEMBL database and was originally curated by [OSB$^+$18]; it consists of 4084 data points in 410 dimensions; we split this into a training set of 3084 points and a test set of 1000 points.

Centering We found that centering the data points decreased error noticeably on the drug discovery dataset, while scaling each coordinate to have variance 1 decreased error by a small amount on the synthetic data. To center in the presence of outliers, we used the robust mean estimation algorithm from [DKK+17].

Adding outliers. We devised a method of generating outliers that fools all of the baselines while still inducing high test error. At a high level, the outliers cause ridge regression to output $w = 0$ (so the model always predicts $y = 0$).

If (X, y) are the true data points and responses, this can be achieved by setting each outlier point $(X_{\text{bad}}, y_{\text{bad}})$ as

$$X_{\text{bad}} = \frac{1}{\alpha \cdot n_{\text{bad}}} y^\top X \quad \text{and} \quad y_{\text{bad}} = -\beta \ ,$$

where n_{bad} is the number of outliers we add, and α and β are hyperparameters.

If $\alpha = \beta$, one can check that $w = 0$ is the unique minimizer for ridge regression on the perturbed dataset. By tuning α and β, we can then obtain attacks that fool all the baselines while damaging the model (we tune α and β separately to give an additional degree of freedom to the attack). To increase the error, we also found it useful to perturb each individual X_{bad} by a small amount of Gaussian noise.

In our experiments we found that this method generated successful attacks as long as the fraction of outliers was at least roughly 2% for synthetic data, and roughly 5% for the drug discovery data.

Results. In Figure 7-2 we compare the test error of our defense against the baselines as we increase the fraction ε of added outliers. To avoid cluttering the figure, we only show the performance of **l2**, **loss**, **gradientCentered**, and SEVER; the performance of the remaining baselines is qualitatively similar to the baselines in Figure 7-2.

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

When the attack parameters α and β are tuned to defeat the baselines (Figure 7-2 left and center), our defense substantially outperforms the baselines as soon as we

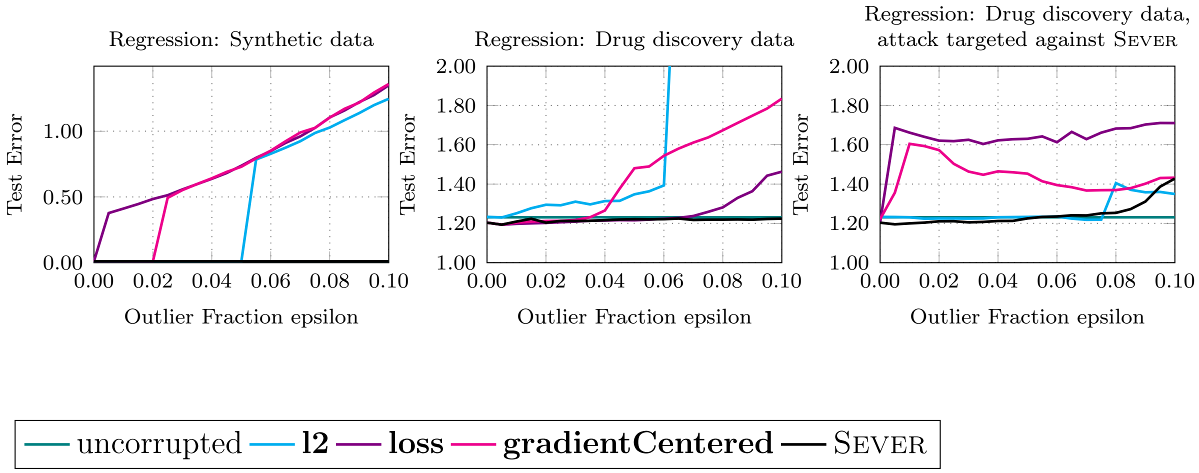


Figure 7-2: ε vs test error for baselines and SEVER on synthetic data and the drug discovery dataset. The left and middle figures show that SEVER continues to maintain statistical accuracy against our attacks which are able to defeat previous baselines. The right figure shows an attack with parameters chosen to increase the test error SEVER on the drug discovery dataset as much as possible. Despite this, SEVER still has relatively small test error.

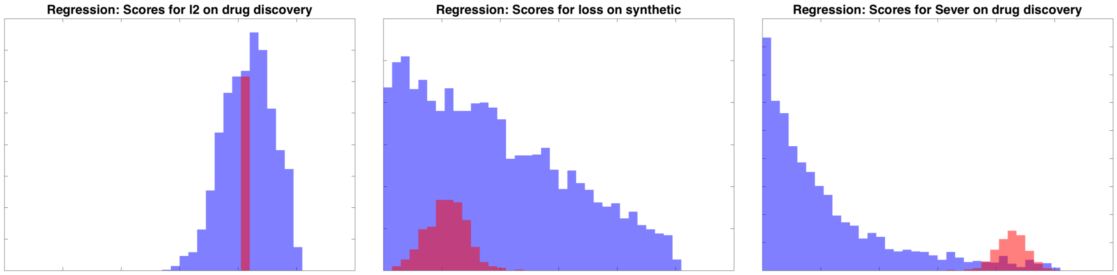


Figure 7-3: A representative set of histograms of scores for baselines and SEVER on synthetic data and a drug discovery dataset. From left to right: scores for the **l2** defense on the drug discovery dataset, scores for **loss** on synthetic data, and scores for SEVER on the drug discovery dataset, all with the addition of 10% outliers. The scores for the true dataset are in blue, and the scores for the outliers are in red. For the baselines, the scores for the outliers are inside the bulk of the distribution and thus hard to detect, whereas the scores for the outliers assigned by SEVER are clearly within the tail of the distribution and easily detectable.

cross $\varepsilon \approx 1.5\%$ for synthetic data, and $\varepsilon \approx 5.5\%$ for the drug discovery data. In fact, most of the baselines do worse than not removing any outliers at all (this is because they end up mostly removing good data points, which causes the outliers to have a larger effect). Even when α and β are instead tuned to defeat SEVER, its resulting error remains small (Figure 7-2 right).

To understand why the baselines fail to detect the outliers, in Figure 7-3 we show a representative sample of the histograms of scores of the uncorrupted points overlaid with the scores of the outliers, for both synthetic data and the drug discovery dataset with $\varepsilon = 0.1$, after one run of the base learner. The scores of the outliers lie well within the distribution of scores of the uncorrupted points. Thus, it would be impossible for the baselines to remove them without also removing a large fraction of uncorrupted points.

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

7.5.2 Support vector machines

We next describe our experimental results for SVMs; we tested our method on a synthetic Gaussian dataset as well as a spam classification task. Similarly to before, the synthetic data consists of observations (x_i, y_i), where $x_i \in \mathbb{R}^{500}$ has independent standard Gaussian entries, and $y_i = \operatorname{sign}(\langle x_i, w^* \rangle + 0.1 z_i)$, where z_i is also Gaussian and w^* is the true parameters (drawn at random from the unit sphere). The spam dataset comes from the Enron corpus [MAP06], and consists of 4137 training points and 1035 test points in 5116 dimensions.

To generate attacks, we used the data poisoning algorithm presented in [SKL17]; the authors provided us with an improved version of their algorithm that can circumvent the **l2** and **loss** baselines and partially circumvents the gradient baselines as well.

In contrast to ridge regression, we did not perform centering and rescaling for these datasets as it did not seem to have a large effect on results.

In all experiments for this section, each method removed the top $p = \frac{n_- + n_+}{\min\{n_+, n_-\}} \cdot \frac{\varepsilon}{r}$ of highest-scoring points for each of $r = 2$ iterations, where n_+ and n_- are the number of positive and negative training points respectively. This expression for p is chosen in order to account for class imbalance, which is extreme in the case of the Enron dataset – if the attacker plants all the outliers in the smaller class, then a smaller value of p would remove too few points, even with a perfect detection method.

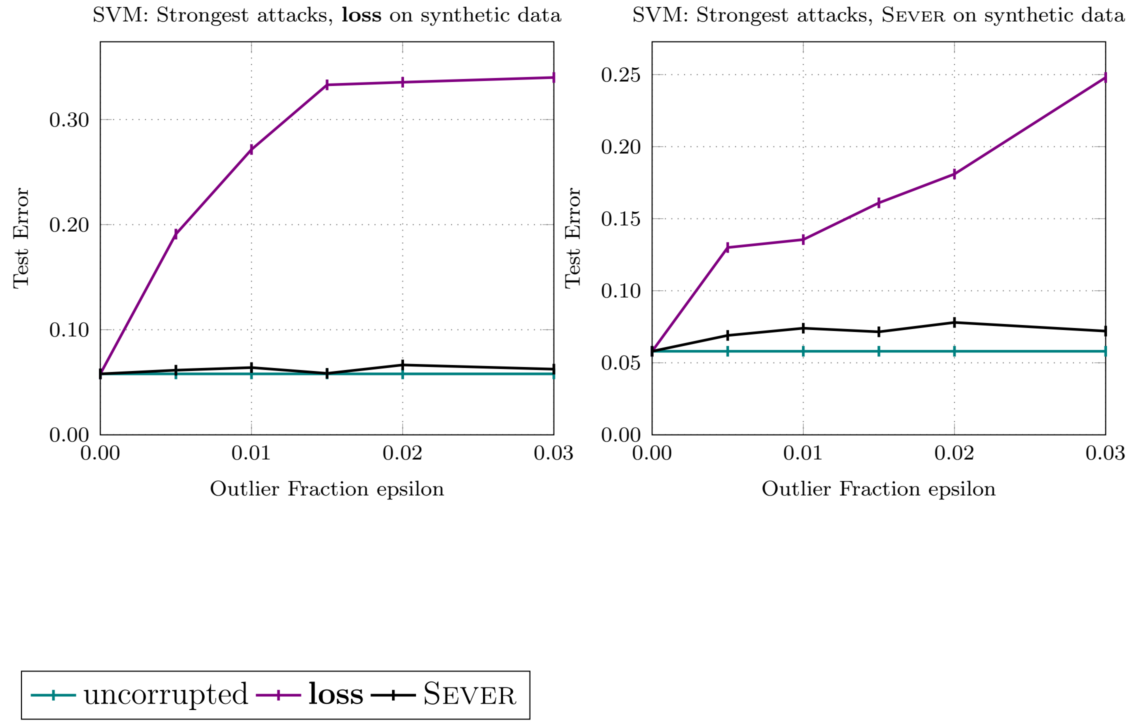


Figure 7-4: ε versus test error for **loss** baseline and SEVER on synthetic data. The left figure demonstrates that SEVER is accurate when outliers manage to defeat **loss**. The right figure shows the result of attacks which increased the test error the most against SEVER. Even in this case, SEVER performs much better than the baselines.

Synthetic results. We considered fractions of outliers ranging from $\varepsilon = 0.005$ to $\varepsilon = 0.03$. By performing a sweep across hyperparameters of the attack, we generated 56 distinct sets of attacks for each value of ε. In Figure 7-4, we show results for the attack where the **loss** baselines does the worst, as well as for the attack where our method does the worst. When attacks are most effective against **loss**, SEVER substantially outperforms it, nearly matching the test accuracy of 5.8% on the uncorrupted data, while **loss** performs worse than 30% error at just a 1.5% fraction of injected outliers. Even when attacks are most effective against SEVER, it still outperforms **loss**, achieving a test error of at most 9.05%. We note that other baselines behaved qualitatively similarly to **loss**, and the results are displayed in Section G.

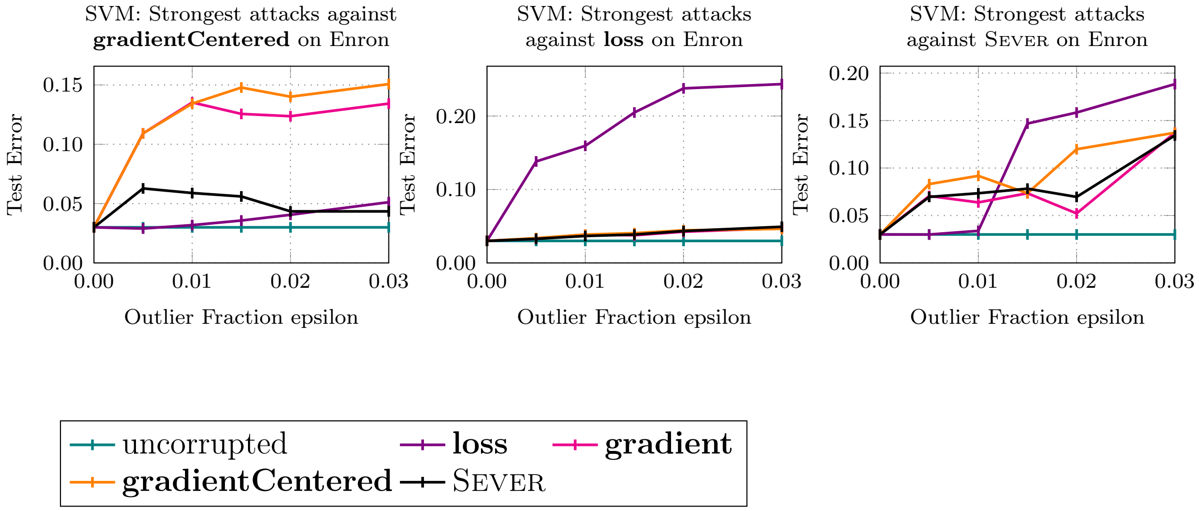


Figure 7-5: ε versus test error for baselines and Sever on the Enron spam corpus. The left and middle figures are the attacks which perform best against two baselines, while the right figure performs best against Sever. Though other baselines may perform well in certain cases, only Sever is consistently accurate. The exception is for certain attacks at $\varepsilon = 0.03$, which, as shown in Figure 7-6, require three rounds of outlier removal for any method to obtain reasonable test error – in these plots, our defenses perform only two rounds.

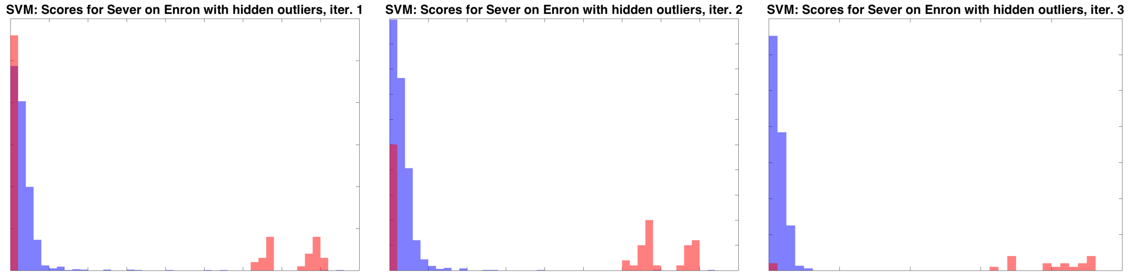


Figure 7-6: An illustration of why multiple rounds of filtering are necessary. Histograms of scores assigned by Sever in three subsequent iterations of outlier removal. Inliers are blue, and outliers are red (scaled up by a factor of 10). In early iterations, a significant fraction of outliers may be "hidden" (i.e. have 0 loss) by being correctly classified in one iteration. However, once previous outliers are removed, these points may become incorrectly classified, thus significantly degrading the quality of our solution but simultaneously becoming evident to Sever.

Spam results. For results on Enron, we used the same values of ε, and considered 96 distinct hyperparameters for the attack. There was not a single attack that simultaneously defeated all of the baselines, so in Figure 7-5 we show two attacks that do well against different sets of baselines, as well as the attack that performs best against our method.

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

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

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

7.6 Discussion

In this paper we have presented an algorithm, Sever, that has both strong theoretical robustness properties in the presence of outliers, and performs well on real datasets. Sever is based on the idea that learning can often be cast as the problem of finding an approximate stationary point of the loss, which can in turn be cast as a robust mean estimation problem, allowing us to leverage existing techniques for efficient robust mean estimation.

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

More general assumptions. The main underlying assumption on which SEVER rests is that the top singular value of the gradients of the data is small. While this appeared to hold true on the datasets we considered, a common occurence in practice is for there to be *a few* large singular values, together with *many* small singular values. It would therefore be desirable to design a version of SEVER that can take advantage of such phenomena. In addition, it would be worthwhile to do a more detailed empirical analysis across a wide variety of datasets investigating properties that can enable robust estimation (the notion of *resilience* in [SCV18] could provide a template for finding such properties).

Stronger robustness to outliers. In theory, SEVER has a $O(\sqrt{\varepsilon})$ dependence in error on the fraction ε of outliers (see Theorem 7.2.1). While without stronger assumptions this is likely not possible to improve, in practice we would prefer to have a dependence closer to $O(\varepsilon)$. Therefore, it would also be useful to improve SEVER to have such an $O(\varepsilon)$-dependence under stronger but realistic assumptions. Unfortunately, all existing algorithms for robust mean estimation that achieve error better than $O(\sqrt{\varepsilon})$ either rely on strong distributional assumptions such as Gaussianity, or else require expensive computation involving like sum-of-squares optimization. Improving the robustness of SEVER thus requires improvements on the robust mean estimation algorithm that SEVER uses as a primitive.

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

It would be interesting to try to learn a representation with such a property. This could be done, for instance, by training a neural network with some cost function that encourages independent features (some ideas along these general lines are discussed in [Ben17]). An issue is how to learn such a representation robustly; one idea is learn a representation on a dataset that is known to be free of outliers, and hope that the representation is useful on other datasets in the same application domain.

Beyond these specific questions, we view the general investigation of robust methods (both empirically and theoretically) as an important step as machine learning moves forwards. Indeed, as machine learning is applied in increasingly many situations and in increasingly automated ways, it is important to attend to robustness considerations so that machine learning systems behave reliably and avoid costly errors. While the bulk of recent work has highlighted the vulnerabilities of machine learning (e.g. [SZS$^+$14, LWSV16, SKL17, EEF$^+$17a, CLL$^+$17a]), we are optimistic that practical algorithms backed by principled theory can finally patch these vulnerabilities and lead to truly reliable systems.

Bibliography

[AAZL18] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *arXiv preprint arXiv:1803.08917*, 2018.

[ABC$^+$18] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. *arXiv preprint arXiv:1802.04633*, 2018.

[ABG$^+$14] Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James R. Voss. The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures. In *COLT*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 1135–1164. JMLR.org, 2014.

[ABK$^+$18] Dan Alistarh, Trevor Brown, Justin Kopinsky, Jerry Li, and Giorgi Nadiradze. Distributionally linearizable data structures. In *SPAA*, 2018.

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

[ACCD11] Ery Arias-Castro, Emmanuel J Candes, and Arnaud Durand. Detection of an anomalous cluster in a network. *The Annals of Statistics*, pages 278–304, 2011.

[ADH$^+$15] Jayadev Acharya, Ilias Diakonikolas, Chinmay Hegde, Jerry Li, and Ludwig Schmidt. Fast and near-optimal algorithms for approximating distributions by histograms. In *PODS*. ACM, 2015.

[ADLS16] Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Fast algorithms for segmented regression. In *ICML*, 2016.

[ADLS17] Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Sample-optimal density estimation in nearly-linear time. In *SODA*, 2017.

[AGL$^+$17] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *NIPS*, 2017.

[AK01] S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. In *STOC*, pages 247–257, 2001.

[AK05] Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical Gaussians. *Ann. Appl. Probab.*, 15(1A):69–92, 2005.

[AKLN17] Dan Alistarh, Justin Kopinsky, Jerry Li, and Giorgi Nadiradze. The power of choice in priority scheduling. In *PODC*. ACM, 2017.

[AKLS15] Dan Alistarh, Justin Kopinsky, Jerry Li, and Nir Shavit. The spraylist: A scalable relaxed priority queue. In *ACM SIGPLAN Notices*, volume 50, pages 11–20. ACM, 2015.

[AM05] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *International Conference on Computational Learning Theory*, pages 458–469. Springer, 2005.

[Api16] Apink. The wave, 2016. Plan A Entertainment.

[AS12] Pranjal Awasthi and Or Sheffet. Improved spectral-norm bounds for clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 37–49. Springer, 2012.

[AW09] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *The Annals of Statistics*, 37(5B):2877–2921, 2009.

[AZL16] Zeyuan Allen-Zhu and Yuanzhi Li. Lazysvd: Even faster svd decomposition yet without agonizing pain. In *NIPS*, 2016.

[BBBB72] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical Inference under Order Restrictions*. Wiley, New York, 1972.

[BCMV14] Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *STOC*, pages 594–603. ACM, 2014.

[BDLS17] Sivaraman Balakrishnan, Simon S Du, Jerry Li, and Aarti Singh. Computationally efficient robust sparse estimation in high dimensions. In *Conference on Learning Theory (COLT)*, 2017.

[Ben17] Y. Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

[Ber41] Andrew C Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

[BGM$^+$16] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *STOC*, 2016.

[BJNP13] Aharon Birnbaum, Iain M Johnstone, Boaz Nadler, and Debashis Paul. Minimax bounds for sparse PCA with noisy high-dimensional data. *Annals of statistics*, 41(3):1055, 2013.

[BKS14] Boaz Barak, Jonathan A. Kelner, and David Steurer. Rounding sum-of-squares relaxations. In *STOC*, pages 31–40. ACM, 2014.

[BKS15] Boaz Barak, Jonathan A. Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *STOC*, pages 143–151. ACM, 2015.

[BLA16] BLAƆKPINK. Playing with fire, 2016. Y.G. Entertainment.

[BM16] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *COLT*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 417–445. JMLR.org, 2016.

[BMV$^+$18] Jess Banks, Cristopher Moore, Roman Vershynin, Nicolas Verzelen, and Jiaming Xu. Information-theoretic bounds and phase transitions in clustering, sparse PCA, and submatrix localization. *IEEE Transactions on Information Theory*, 2018.

[BNJT10] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

[BNL12] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, pages 1467–1474, 2012.

[BR13] Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse pca. *arXiv preprint arXiv:1304.0828*, 2013.

[Bru09] S. C. Brubaker. *Extensions of Principle Components Analysis*. PhD thesis, Georgia Institute of Technology, 2009.

[BS10a] M. Belkin and K. Sinha. Polynomial learning of distribution families. In *FOCS*, 2010.

[BS10b] Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *FOCS*, pages 103–112. IEEE Computer Society, 2010.

[BS14] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *CoRR*, abs/1404.5236, 2014.

[BS17] Boaz Barak and David Steurer. The sos algorithm over general domains. `http://www.sumofsquares.org/public/lec-definitions-general.html`, 2017. [Online; accessed 11-1-2017].

[BTS18a] BTS. D.N.A., 2018. Bighit Entertainment.

[BTS18b] BTS. The truth untold, 2018. Bighit Entertainment.

[BV08] S. C. Brubaker and S. Vempala. Isotropic PCA and affine-invariant clustering. In *FOCS*, 2008.

[BWY14] Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *CoRR*, abs/1408.2156, 2014.

[CDSS13] S. Chan, I. Diakonikolas, R. Servedio, and X. Sun. Learning mixtures of structured distributions over discrete domains. In *SODA*, pages 1380–1394, 2013.

[CDSS14a] S. Chan, I. Diakonikolas, R. Servedio, and X. Sun. Efficient density estimation via piecewise polynomial approximation. In *STOC*, pages 604–613, 2014.

[CDSS14b] S. Chan, I. Diakonikolas, R. Servedio, and X. Sun. Near-optimal density estimation in near-linear time using variable-width histograms. In *NIPS*, pages 1844–1852, 2014.

[CGG02] M. Cryan, L. Goldberg, and P. Goldberg. Evolutionary trees can be learned in polynomial time in the two state general Markov model. *SIAM Journal on Computing*, 31(2):375–397, 2002.

[CLL$^+$17a] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[CLL$^+$17b] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[CLMW11] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

[CMV$^+$16] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security)*, pages 513–530, 2016.

[CMW$^+$13] T Tony Cai, Zongming Ma, Yihong Wu, et al. Sparse pca: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.

[CR09] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[CRPW12] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[CRZ^{+}16] T Tony Cai, Zhao Ren, Harrison H Zhou, et al. Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation. *Electronic Journal of Statistics*, 10(1):1–59, 2016.

[CSV17] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Symposium on Theory of Computing*, 2017.

[CT06] Thomas M Cover and Joy A Thomas. Elements of information theory 2nd edition. 2006.

[CW01] A Carbery and J Wright. Distributional and lˆ q norm inequalities for polynomials over convex bodies in rˆ n. *Mathematical research letters*, 8(3):233–248, 2001.

[CW08] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[D^{+}14] Jeff Donahue et al. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[Das99] Sanjoy Dasgupta. Learning mixtures of gaussians. In *Foundations of computer science, 1999. 40th annual symposium on*, pages 634–644. IEEE, 1999.

[dBG08] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294, 2008.

[DBS17] Simon S Du, Sivaraman Balakrishnan, and Aarti Singh. Computationally efficient robust estimation of sparse functionals. *arXiv preprint arXiv:1702.07709*, 2017.

[DDKT16] C. Daskalakis, A. De, G. Kamath, and C. Tzamos. A size-free CLT for Poisson multinomials and its applications. In *Proceedings of the 48th Annual ACM Symposium on the Theory of Computing*, STOC, 2016.

[DDO^{+}13] C. Daskalakis, I. Diakonikolas, R. O'Donnell, R.A. Servedio, and L. Tan. Learning Sums of Independent Integer Random Variables. In *FOCS*, pages 217–226, 2013.

[DDS12] C. Daskalakis, I. Diakonikolas, and R.A. Servedio. Learning k-modal distributions via testing. In *SODA*, pages 1371–1385, 2012.

[DDS15] A. De, I. Diakonikolas, and R. Servedio. Learning from satisfying assignments. In *SODA*, 2015.

[dEGJL07] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I Jordan, and Gert RG Lanckriet. A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM Review*, 49(3):434–448, 2007.

[DG85] L. Devroye and L. Györfi. *Nonparametric Density Estimation: The L_1 View.* John Wiley & Sons, 1985.

[DGL$^+$17] Ilias Diakonikolas, Elena Grigorescu, Jerry Li, Abhiram Natarajan, Krzysztof Onak, and Ludwig Schmidt. Communication-efficient distributed learning of discrete distributions. In *NIPS*, 2017.

[DK14] Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of gaussians. In *Conference on Learning Theory*, 2014.

[DKK$^+$16] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS)*, pages 655–664. IEEE, 2016.

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

[DKK$^+$18a] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a gaussian: Getting optimal error, efficiently. In *Symposium on Discrete Algorithms (SODA)*, 2018.

[DKK$^+$18b] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. *arXiv preprint arXiv:1803.02815*, 2018.

[DKS16a] I. Diakonikolas, D. M. Kane, and A. Stewart. The fourier transform of Poisson multinomial distributions and its algorithmic applications. In *STOC*, 2016.

[DKS16b] I. Diakonikolas, D. M. Kane, and A. Stewart. Optimal learning via the Fourier transform for sums of independent integer random variables. In *Proceedings of the 29th Annual Conference on Learning Theory*, COLT, pages 831–849, 2016.

[DKS16c] I. Diakonikolas, D. M. Kane, and A. Stewart. Robust learning of fixed-structure Bayesian networks. *CoRR*, abs/1606.07384, 2016.

[DKS16d] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. *arXiv preprint arXiv:1611.03473*, 2016.

[DKS17] I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '17, pages 73–84, Washington, DC, USA, 2017. IEEE Computer Society.

[DKS18a] I. Diakonikolas, D. M. Kane, and A. Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, New York, NY, USA, 2018. ACM.

[DKS18b] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Symposium on Theory of Computing (STOC)*, 2018.

[DKW56] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

[DL12] Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation.* Springer Science & Business Media, 2012.

[DLS18] Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Fast and sample near-optimal algorithms for learning multidimensional histograms. In *COLT*, 2018.

[DS07] Sanjoy Dasgupta and Leonard Schulman. A probabilistic analysis of em for mixtures of separated, spherical gaussians. *Journal of Machine Learning Research*, 8(Feb):203–226, 2007.

[DTZ17] Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of em suffice for mixtures of two gaussians. *Conference on Learning Theory*, 2017.

[Dur10] Rick Durrett. *Probability: theory and examples.* Cambridge university press, 2010.

[EEF$^+$17a] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. Robust physical-world attacks on machine learning models. *arXiv*, 2017.

[EEF$^+$17b] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945*, 2017.

[ER15] Robert Epstein and Ronald E Robertson. The search engine manipulation effect (seme) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, 112(33):E4512–E4521, 2015.

[FB87] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

[FM99] Y. Freund and Y. Mansour. Estimating a mixture of two product distributions. In *COLT*, pages 183–192, 1999.

[FSO06] Jon Feldman, Rocco A. Servedio, and Ryan O'Donnell. PAC learning axis-aligned mixtures of gaussians with no separation assumption. In *COLT*, volume 4005 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2006.

[GDGG17] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[GGLS14] Rati Gelashvili, Mohsen Ghaffari, Jerry Li, and Nir Shavit. On the importance of registers for computability. In *OPODIS*. Springer, 2014.

[GHK15] Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of Gaussians in high dimensions [extended abstract]. In *STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing*, pages 761–770. ACM, New York, 2015.

[Gir07] Girls' Generation. Into the new world, 2007. S.M. Entertainment.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer, 1988.

[GM15] Rong Ge and Tengyu Ma. Decomposing overcomplete 3rd order tensors using sum-of-squares algorithms. *arXiv preprint arXiv:1504.05287*, 2015.

[GMN14] Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of distributed statistical estimation and dimensionality. In *NIPS*, 2014.

[GSS14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.

[Gur16] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016.

[GWL14] Quanquan Gu, Zhaoran Wang, and Han Liu. Sparse PCA with oracle property. In *NIPS*, 2014.

[HK13] Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *ITCS'13—Proceedings of the 2013 ACM Conference on Innovations in Theoretical Computer Science*, pages 11–19. ACM, New York, 2013.

[HKP$^+$17] Samuel B Hopkins, Pravesh Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. *Symposium on Foundations of Computer Science*, 2017.

[HL18] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Symposium on Theory of Computing (STOC)*, 2018.

[HP15a] M. Hardt and E. Price. Sharp bounds for learning a mixture of two Gaussians. In *STOC*, 2015.

[HP15b] Moritz Hardt and Eric Price. Tight bounds for learning a mixture of two gaussians. In *STOC*, pages 753–760. ACM, 2015.

[HSS15] Samuel B. Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *COLT*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 956–1006. JMLR.org, 2015.

[HSSS16] Samuel B. Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. In *STOC*, pages 178–191. ACM, 2016.

[HTW15] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[Hub64] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[Hub97] P. J. Huber. Robustness: Where are we now? *Lecture Notes-Monograph Series*, pages 487–498, 1997.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[IKO18] IKON. Love scenario, 2018. Y.G. Entertainment.

[Im17] Yoona Im. When the wind blows, 2017. S.M. Entertainment.

[IOI16] IOI. Downpour, 2016. CJ E&M.

[IOI17] IOI. When the cherry blossoms fade, 2017. CJ E&M.

[JNRS10] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(Feb):517–553, 2010.

[Joh01] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.

[Joh13] Iain M Johnstone. Gaussian estimation: Sequence and wavelet models, 2013. unpublished manuscript, available at `http://statweb.stanford.edu/~imj/GE06-11-13.pdf`.

[KGB16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[KH09] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[KK10] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *FOCS*, pages 299–308. IEEE Computer Society, 2010.

[KKM18] A. Klivans, P. K. Kothari, and R. Meka. Efficient algorithms for outlier-robust regression. *CoRR*, abs/1803.03241, 2018.

[KL93] M. J. Kearns and M. Li. Learning in the presence of malicious errors. *SICOMP*, 22(4):807–837, 1993.

[KL17] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

[KLS09] A. Klivans, P. Long, and R. Servedio. Learning halfspaces with malicious noise. In *ICALP*, 2009.

[KLSU18] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. *arXiv preprint arXiv:1805.00216*, 2018.

[KMR$^+$94] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proc. 26th STOC*, pages 273–282, 1994.

[KMV10] A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two Gaussians. In *STOC*, pages 553–562, 2010.

[KMY$^+$16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[KNV$^+$15] Robert Krauthgamer, Boaz Nadler, Dan Vilenchik, et al. Do semidefinite relaxations solve sparse PCA up to the information limit? *The Annals of Statistics*, 43(3):1300–1322, 2015.

[KS18] Pravesh K Kothari and Jacob Steinhardt. Better agnostic clustering via relaxed tensor norms. In *Symposium on Theory of Computing (STOC)*, 2018.

[LAT$^+$08] J.Z. Li, D.M. Absher, H. Tang, A.M. Southwick, A.M. Casto, S. Ramachandran, H.M. Cann, G.S. Barsh, M. Feldman, L.L. Cavalli-Sforza, and R.M. Myers. Worldwide human relationships inferred from genome-wide patterns of variation. *Science*, 319:1100–1104, 2008.

[Lee17] Jieun Lee. Through the night, 2017. Fave Entertainment.

[Li17] Jerry Li. Robust sparse estimation tasks in high dimensions. *arXiv preprint arXiv:1702.05860*, 2017.

[LM00] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[LMPS18] Jerry Li, Aleksander Mądry, John Peebles, and Ludwig Schmidt. On the limitations of first order approximation in GAN dynamics. In *ICML*, 2018.

[LMTZ15] Gilad Lerman, Michael B McCoy, Joel A Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

[Löf04] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *CACSD*, 2004.

[LP15] Jerry Li and John Peebles. Replacing mark bits with randomness in fibonacci heaps. In *ICALP*. Springer, 2015.

[LRV16] K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *FOCS*, 2016.

[LS17] Jerry Li and Ludwig Schmidt. Robust and proper learning for mixtures of gaussians via systems of polynomial inequalities. In *COLT*, 2017.

[LWSV16] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[LZ12] Zhaosong Lu and Yong Zhang. An augmented lagrangian approach for sparse principal component analysis. *Mathematical Programming*, 135(1-2):149–193, 2012.

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

[MAP06] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive Bayes – which naive Bayes? In *CEAS*, volume 17, pages 28–69, 2006.

[MM15] Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *NIPS*, 2015.

[MMS$^+$17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[MP04] Geoffrey McLachlan and David Peel. *Finite mixture models.* John Wiley & Sons, 2004.

[MR05] E. Mossel and S. Roch. Learning nonsingular phylogenies and Hidden Markov Models. In *STOC*, 2005.

[MSS16] Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *FOCS*, pages 438–446. IEEE Computer Society, 2016.

[MV10] Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *FOCS*, pages 93–102. IEEE Computer Society, 2010.

[MV17] M. Meister and G. Valiant. A data prism: Semi-verified learning in the small-alpha regime. *CoRR*, abs/1708.02740, 2017.

[MVW17] Dustin G Mixon, Soledad Villar, and Rachel Ward. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, page iax001, 2017.

[MW15] Tengyu Ma and Avi Wigderson. Sum-of-squares lower bounds for sparse pca. In *NIPS*, 2015.

[NJB$^+$08] J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, A. Auton, A. Indap, K. S. King, S. Bergmann, M. R. Nelson, et al. Genes mirror geography within europe. *Nature*, 456(7218):98–101, 2008.

[NPXNR14] A. Newell, R. Potharaju, L. Xiang, and C. Nita-Rotaru. On the practicality of integrity attacks on document-level sentiment analysis. In *Workshop on Artificial Intelligence and Security (AISec)*, pages 83–93, 2014.

[O'D17] Ryan O'Donnell. Sos is not obviously automatizable, even approximately. 2017.

[OMH$^+$14] Alexei Onatski, Marcelo J Moreira, Marc Hallin, et al. Signal detection in high dimension: The multispiked case. *The Annals of Statistics*, 42(1):225–254, 2014.

[OSB$^+$18] I. Olier, N. Sadawi, G. R. Bickerton, J. Vanschoren, C. Grosan, L. Soldatova, and Ross D. King. Meta-qsar: a large-scale application of meta-learning to drug design and discovery. *Machine Learning*, 107(1):285–311, Jan 2018.

[OZ13] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *SODA*, pages 1537–1556. SIAM, 2013.

[PCG$^+$16] Nicolas Papernot, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Fartash Faghri, Alexander Matyasko, Karen Hambardzumyan, Yi-Lin Juang, Alexey Kurakin, Ryan Sheatsley, et al. cleverhans v2. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.

[Pea94] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[PLJD10] P. Paschou, J. Lewis, A. Javed, and P. Drineas. Ancestry informative markers for fine-scale individual assignment to worldwide populations. *Journal of Medical Genetics*, 47:835–847, 2010.

[PS17] Aaron Potechin and David Steurer. Exact tensor completion with sum-of-squares. *CoRR*, abs/1702.06237, 2017.

[PSBR18] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust estimation via robust gradient estimation. *CoRR*, abs/1802.06485, 2018.

[PWBM16] Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Optimality and sub-optimality of PCA for spiked random matrices and synchronization. *arXiv preprint arXiv:1609.05573*, 2016.

[QV18] M. Qiao and G. Valiant. Learning discrete distributions from untrusted batches. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, pages 47:1–47:20, New York, NY, USA, 2018. ACM.

[Red14] Red Velvet. Happiness, 2014. S.M. Entertainment.

[Red15] Red Velvet. Dumb Dumb, 2015. S.M. Entertainment.

[Red17a] Red Velvet. Peekaboo, 2017. S.M. Entertainment.

[Red17b] Red Velvet. Red Flavor, 2017. S.M. Entertainment.

[RH17] Philippe Rigollet and Jan-Christian Hütter. *High Dimensional Statistics*. 2017.

[RPW$^+$02] N. Rosenberg, J. Pritchard, J. Weber, H. Cann, K. Kidd, L.A. Zhivotovsky, and M.W. Feldman. Genetic structure of human populations. *Science*, 298:2381–2385, 2002.

[RV17] Oded Regev and Aravindan Vijayraghavan. On learning mixtures of well-separated gaussians. In *Symposium on Foundations of Computer Science*, 2017.

[RW17] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. *CoRR*, abs/1702.05139, 2017.

[SBBR16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016.

[Sco92] D.W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley, New York, 1992.

[SCV18] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Innovations in Theoretical Computer Science (ITCS)*, 2018.

[SD15] Jacob Steinhardt and John Duchi. Minimax rates for memory-bounded sparse linear regression. In *COLT*, 2015.

[Ser03] R. Servedio. Smooth boosting and learning with malicious noise. *JMLR*, 4:633–648, 2003.

[Sha16] Ohad Shamir. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. In *ICML*, 2016.

[SHN$^+$18] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

[Sil86] B. W. Silverman. *Density Estimation*. Chapman and Hall, London, 1986.

[SKL17] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *NIPS*, 2017.

[SOAJ14] Ananda Theertha Suresh, Alon Orlitsky, Jayadev Acharya, and Ashkan Jafarpour. Near-optimal-sample estimators for spherical gaussian mixtures. In *Advances in Neural Information Processing Systems*, pages 1395–1403, 2014.

[SS17] Tselil Schramm and David Steurer. Fast and robust tensor decomposition with applications to dictionary learning. *Conference on Learning Theory*, 2017.

[SZS$^+$14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[TKP$^+$17] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[TLM18] Brandon Tran, Jerry Li, and Aleksander Mądry. Spectral signatures in backdoor attacks for neural networks. 2018.

[Tro12] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[Tro15] J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

[TSM85] D Michael Titterington, Adrian FM Smith, and Udi E Makov. *Statistical analysis of finite mixture distributions*. Wiley,, 1985.

[Tsy08] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

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

[Twi15] Twice. Like a fool, 2015. J.Y.P. Entertainment.

[Twi17] Twice. Signal, 2017. J.Y.P. Entertainment.

[Val85] L. Valiant. Learning disjunctions of conjunctions. In *IJCAI*, pages 560–566, 1985.

[Ver10] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[VW02] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixtures of distributions. In *FOCS*, page 113. IEEE Computer Society, 2002.

[WBS16] Tengyao Wang, Quentin Berthet, and Richard J Samworth. Statistical and computational trade-offs in estimation of sparse principal components. *The Annals of Statistics*, 2016.

[Web29] Alfred Weber. *Theory of the Location of Industries*. University of Chicago Press, 1929.

[WGL16] Zhaoran Wang, Quanquan Gu, and Han Liu. On the statistical limits of convex relaxations. In *ICML*, 2016.

[WTH09] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[Wu83] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[XCS10] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[XHM16] Ji Xu, Daniel J Hsu, and Arian Maleki. Global analysis of expectation maximization for mixtures of two gaussians. In *Advances in Neural Information Processing Systems*, pages 2676–2684, 2016.

[ZL14] T. Zhang and G. Lerman. A novel m-estimator for robust pca. *J. Mach. Learn. Res.*, 15(1):749–808, January 2014.

[ZLK$^+$17] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. The zipml framework for training models with end-to-end low precision: The cans, the cannots, and a little bit of deep learning. In *ICML 2017*, 2017.

Just like words in the sand, written in the domain of the waves,
I fear that you will soon disappear to a faraway place.
Please know that I will always miss you.
Always.

Appendix A

Omitted Proofs from Chapter 1

A.1 Omitted Proofs from Section 4.3

A.1.1 Proof of Fact 1.4.3

Proof of Fact 1.4.3. Observe that by rotational and translational invariance, it suffices to consider the problem when $\mu_1 = -\varepsilon e_1/2$ and $\mu_2 = \varepsilon e_1/2$, where e_1 is the first standard basis vector. By the decomposability of TV distance, we have that the TV distance can in fact be written as a 1 dimensional integral:

$$d_{\mathrm{TV}}\left(\mathcal{N}(\mu_1, I), \mathcal{N}(\mu_2, I)\right) = \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left| e^{-(x-\varepsilon/2)^2/2} - e^{-(x+\varepsilon/2)^2/2} \right| dx \; .$$

The value of the function $f(x) = e^{-(x-\varepsilon/2)^2/2} - e^{-(x+\varepsilon/2)^2/2}$ is negative when $x < 0$ and positive when $x > 0$, hence this integral becomes

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{N}(\mu_1, I), \mathcal{N}(\mu_2, I)\right) &= \frac{1}{\sqrt{2\pi}} \int_{0}^{\infty} e^{-(x-\varepsilon/2)^2/2} - e^{-(x+\varepsilon/2)^2/2} dx \\ &= F(\varepsilon/2) - F(-\varepsilon/2) \; , \end{aligned}$$

where $F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$ is the CDF of the standard normal Gaussian. By Taylor's theorem, and since $F''(x)$ is bounded when $x \in [-1, 1]$, we have

$$
\begin{aligned}
F(\varepsilon/2) - F(-\varepsilon/2) &= F'(-\varepsilon/2)\varepsilon + O(\varepsilon^3) \\
&= \frac{1}{\sqrt{2\pi}} e^{-(\varepsilon/2)^2/2} \varepsilon + O(\varepsilon^3) \\
&= \left(\frac{1}{\sqrt{2\pi}} + o(1) \right) \varepsilon \ ,
\end{aligned}
$$

which proves the claim. □

A.1.2 Proof of Corollary 1.4.6

Proof of Corollary 1.4.6. Let $M = \Sigma_2^{-1/2} \Sigma_1 \Sigma_2^{-1/2}$. Then (1.1) simplifies to

$$
d_{\mathrm{KL}} \left(\mathcal{N}(\mu_1, \Sigma_1) \| \mathcal{N}(\mu_2, \Sigma_2) \right) = \frac{1}{2} \left(\mathrm{tr}(M) - d - \ln \det(M) \right) \ . \tag{A.1}
$$

Observe that $\|\Sigma_1 - \Sigma_2\|_{\Sigma_2} = \varepsilon$ is equivalent to the statement that $\|I - M\|_F = \varepsilon$.

Since both terms in the last line of (A.1) are rotationally invariant, we may assume without loss of generality that M is diagonal. Let $M = \mathrm{diag}(1 + \lambda_1, \ldots, 1 + \lambda_d)$. Thus, the KL divergence between the two distributions is given exactly by $\frac{1}{2} \sum_{i=1}^{d} \left(\lambda_i - \log(1 + \lambda_i) \right)$, where we are guaranteed that $(\sum_{i=1}^{d} \lambda_i^2)^{1/2} = \varepsilon$. By the second order Taylor approximation to $\ln(1 + x)$, for x small, we have that for ε sufficiently small,

$$
\sum_{i=1}^{d} \lambda_i - \log(1 + \lambda_i) = \Theta \left(\sum_{i=1}^{d} \lambda_i^2 \right) = \Theta(\varepsilon^2) \ .
$$

Thus, we have shown that for ε sufficiently small, $d_{\mathrm{KL}} \left(\mathcal{N}(\mu_1, \Sigma_1) \| \mathcal{N}(\mu_2, \Sigma_2) \right) \leq O(\varepsilon^2)$. The result now follows by an application of Pinsker's inequality (Fact 1.4.4). □

Appendix B

Deferred Proofs from Chapter 2

B.1 Proofs of Concentration Inequalities

This section contains deferred proofs of several concentration inequalities.

Proof of Corollary 2.1.14: Let $\mathfrak{S}_m = \{S \subseteq [n] : |S| = m\}$ denote the set of subsets of $[n]$ of size m. The same Bernstein-style analysis as in the proof of Lemma 2.1.8 yields that there exist universal constants A, B so that:

$$\begin{aligned}\Pr &\left[\exists T \in \mathfrak{S}_m : \left\| \frac{1}{m} \sum_{i \in T} X_i X_I^\top - I \right\|_F \geq O\left(\gamma_2 \frac{n}{m}\right) \right] \\ &\leq 4 \exp\left(\log \binom{n}{m} + Ad^2 - B\gamma_2 n \right) \; .\end{aligned}$$

Thus, union bounding over all $m \in \{1, \ldots, \varepsilon n\}$ yields that

$$\begin{aligned}\Pr &\left[\exists T \text{ s.t.} |T| \leq \varepsilon n : \left\| \frac{1}{|T|} \sum_{i \in T} X_i X_I^\top - I \right\|_F \geq O\left(\gamma_2 \frac{n}{|T|}\right) \right] \\ &\leq 4 \exp\left(\log(\varepsilon n) + \log \binom{n}{\varepsilon n} + Ad^2 - B\gamma_2 n \right) \leq \delta \; ,\end{aligned}$$

by the same manipulations as in the proof of Lemma 2.1.8. □

Proof of Theorem 2.1.15: We first recall Isserlis' theorem, which we will require in this proof.

Theorem B.1.1 (Isserlis' theorem)**.** *Let $a_1, \dots, a_k \in \mathbb{R}^d$ be fixed vectors. Then if $X \sim \mathcal{N}(0, I)$, we have*

$$\mathbb{E}\left[\prod_{i=1}^{k} \langle a_i, X \rangle\right] = \sum \prod \langle a_i, a_j \rangle \ ,$$

where the $\sum \prod$ is over all matchings of $\{1, \dots, k\}$.

Let $v = A^\flat \in \mathcal{S}_{\text{sym}}$. We will show that

$$\langle v, Mv \rangle = 2v^\top \left(\Sigma^{\otimes 2}\right) v + v^\top \left(\Sigma^\flat\right) \left(\Sigma^\flat\right)^\top v \ .$$

Since M is a symmetric operator on $\mathbb{R}^{d^2}$, its quadratic form uniquely identifies it and this suffices to prove the claim.

Since A is symmetric, it has a eigenvalue expansion $A = \sum_{i=1}^d \lambda_i u_i u_i^\top$, which immediately implies that $v = \sum_{i=1}^d \lambda_i u_i \otimes u_i$. Let $X \sim \mathcal{N}(0, \Sigma)$. We compute the quadratic form:

$$\begin{aligned}
\langle v, Mv \rangle &= \sum_{i,j=1}^{d} \lambda_i \lambda_j \langle u_i \otimes u_i, \mathbb{E}[(X \otimes X)(X \otimes X)^\top] u_j \otimes u_j \rangle \\
&= \sum_{i,j=1}^{d} \lambda_i \lambda_j \, \mathbb{E}\left[\langle u_i \otimes u_i, (X \otimes X)(X \otimes X)^\top u_j \otimes u_j \rangle\right] \\
&= \sum_{i,j=1}^{d} \lambda_i \lambda_j \, \mathbb{E}\left[\langle u_i, X \rangle^2 \langle u_j, X \rangle^2\right] \\
&= \sum_{i,j=1}^{d} \lambda_i \lambda_j \, \mathbb{E}\left[\langle B^\top u_i, Y \rangle^2 \langle B^\top u_j, Y \rangle^2\right] \\
&= \sum_{i,j=1}^{d} \lambda_i \lambda_j \left(\langle B^\top u_i, B^\top u_i \rangle \langle B^\top u_j, B^\top u_j \rangle + 2 \langle B^\top u_i, B^\top u_j \rangle^2\right) \ ,
\end{aligned}$$

where the last line follows by invoking Isserlis's theorem. We now manage both sums individually. We have

$$\begin{aligned}\sum_{i,j=1}^{d} \lambda_i\lambda_j \langle B^\top u_i, B^\top u_i\rangle \langle B^\top u_j, B^\top u_j\rangle &= \left(\sum_{i=1}^{d} \lambda_i u_i^\top \Sigma u_i\right)^2 \\ &= \left(\sum_{i=1}^{d} \lambda_i \left(u_i \otimes u_i\right)^\top \left(\Sigma^\flat\right)\right)^2 \\ &= v^\top \left(\Sigma^\flat\right) \left(\Sigma^\flat\right)^\top v \ ,\end{aligned}$$

and

$$\begin{aligned}\sum_{i,j=1}^{d} \lambda_i\lambda_j \langle B^\top u_i, B^\top u_j\rangle^2 &= \sum_{i,j} \lambda_i\lambda_j \langle (B^\top u_i)^{\otimes 2}, (B^\top u_j)^{\otimes 2}\rangle \\ &= \sum_{i,j=1}^{d} \lambda_i\lambda_j \langle (B^\top \otimes B^\top) u_i \otimes u_i, (B^\top \otimes B^\top) u_j \otimes u_j\rangle \\ &= \sum_{i,j=1}^{d} \lambda_i\lambda_j (u_i \otimes u_i) \Sigma^{\otimes 2} (u_j \otimes u_j) \\ &= v^\top \Sigma^{\otimes 2} v \ .\end{aligned}$$

□

B.1.1 Proof of Theorem 2.1.16

This follows immediately from Lemmas 5.4.3 and 5.4.6.

Appendix C

Deferred Proofs from Chapter 3

C.1 Information theoretic estimators for robust sparse estimation

This section is dedicated to the proofs of the following two facts:

Fact C.1.1. *Fix $\varepsilon, \delta > 0$, and let k be fixed. Given an ε-corrupted set of samples $X_1, \ldots, X_n \in \mathbb{R}^d$ from $\mathcal{N}(\mu, I)$, where μ is k-sparse, and*

$$n = O\left(\frac{k \log(d/\varepsilon) + \log 1/\delta}{\varepsilon^2}\right) ,$$

there is an (inefficient) algorithm which outputs $\widehat{\mu}$ so that with probability $1 - \delta$, we have $\|\mu - \widehat{\mu}\|_2 \leq O(\varepsilon)$. Moreover, up to logarithmic factors, this rate is optimal.

Fact C.1.2. *Fix $\rho, \delta > 0$. Suppose that $\rho = O(1)$. Then, there exist universal constants c, C so that: (a) if $\varepsilon \leq c\rho$, and we are given a ε-corrupted set of samples from either $\mathcal{N}(0, I)$ or $\mathcal{N}(0, I + \rho v v^\top)$ for some k-sparse unit vector v of size*

$$n = \Omega\left(\frac{k + \log \binom{d}{k} + \log 1/\delta}{\rho^2}\right) ,$$

then there is an (inefficient) algorithm which succeeds with probability $1 - \delta$ for the detection problem. Moreover, if $\varepsilon \geq C\rho$, then no algorithm succeeds with probability *greater than $1/2$, and this statistical rate is optimal.*

The rates in Facts C.1.1 and C.1.2 are already known to be optimal (up to log factors) without noise. Thus in this section we focus on proving the upper bounds, and the lower bounds on error.

The lower bounds on what error is achievable follow from the following two facts, which follow from Pinsker's inequality (see e.g. [CT06]), and the fact that the corruption model can, given samples from D_1, simulate samples from D_2 by corrupting an $O(\varepsilon)$ fraction of points, if $d_{\mathrm{TV}}(D_1, D_2) \leq O(\varepsilon)$.

Fact C.1.3. *Fix $\varepsilon > 0$ sufficiently small. Let μ_1, μ_2 be arbitrary. There is some universal constant C so that if $d_{\mathrm{TV}}(\mathcal{N}(\mu_1, I), \mu_2, I) \leq \varepsilon$, then $\|\mu_1 - \mu_2\|_2 \leq C\varepsilon$, and if $\|\mu_1 - \mu_2\|_2 \leq \varepsilon$, then $d_{\mathrm{TV}}(\mathcal{N}(\mu_1, I), \mathcal{N}(\mu_2, I)) \leq C\varepsilon$.*

Fact C.1.4. *Fix $\rho = O(1)$. Let u, v be arbitrary unit vectors. Then $d_{\mathrm{TV}}(\mathcal{N}(0, I), \mathcal{N}(0, I + \rho vv^\top)) = \Theta(\rho)$, and $d_{\mathrm{TV}}(\mathcal{N}(0, I + \rho vv^\top), \mathcal{N}(0, I + \rho uu^\top)) = O(L(u, v))$.*

Our techniques for proving the upper bounds go through the technique of agnostic hypothesis selection via tournaments. Specifically, we use the following lemma:

Lemma C.1.5 ([DKK$^+$16], Lemma 2.9)**.** *Let $\mathcal{C}$ be a class of probability distributions. Suppose that for some $n, \varepsilon, \delta > 0$ there exists an algorithm that given an ε-corrupted set of samples from some $D \in \mathcal{C}$, returns a list of M distributions so that with $1 - \delta/3$ probability there exists a $D' \in M$ with $d_{\mathrm{TV}}(D', D) < \gamma$. Suppose furthermore that with probability $1 - \delta/3$, the distributions returned by this algorithm are all in some fixed set $\mathcal{M}$. Then there exists another algorithm, which given $O(N + (\log(|\mathcal{M}|) + \log(1/\delta))/\varepsilon^2)$ samples from Π, an ε-fraction of which have been arbitrarily corrupted, returns a single distribution Π' so that with $1 - \delta$ probability $d_{\mathrm{TV}}(D', D) < O(\gamma + \varepsilon)$.*

C.1.1 Proof of Upper Bound in Fact C.1.1

Let $\mathcal{M}_A$ be the set of distributions $\{\mathcal{N}(\mu', I)\}$, where μ' ranges over the set of k-sparse vectors so that each coordinate of μ' is an integer multiple of $\varepsilon/(10\sqrt{d})$, and so that $\|\mu' - \mu\|_2 \leq A$. We then have:

Claim C.1.6. *There exists a $\mathcal{N}(\mu', I) = D \in \mathcal{M}_A$ so that $\|\mu - \mu'\|_2 \leq O(\varepsilon)$. Moreover, $|\mathcal{M}_A| \leq \binom{d}{k} \cdot (10A\sqrt{d}/\varepsilon)^k$.*

Proof. The first claim is straightforward. We now prove the second claim. For each possible set of k coordinates, there are at most $(10A\sqrt{d}/\varepsilon)^k$ vectors supported on those k coordinates with each coordinate being an integer multiple of $\varepsilon/(10\sqrt{d})$ with distance at most A from any fixed vector. Enumerating over all $\binom{d}{k}$ possible choices of k coordinates yields the desired answer. □

The estimator is given as follows: first, run NaivePrune$(X_1, \ldots, X_n, \delta)$ to output some μ_0 so that with probability $1 - \delta$, we have $\|\mu_0 - \mu\|_2 \leq O(\sqrt{d \log n/\delta})$. Round each coordinate of μ_0 so that it is an integer multiple of $\varepsilon/(10\sqrt{d})$. Then, output the set of distributions $\mathcal{M}' = \{\mathcal{N}(\mu'', I)\}$, where μ'' is any k-sparse vector with each coordinate being an integer multiple of $\varepsilon/(10\sqrt{d})$, with $\|\alpha\|_2 \leq O(\sqrt{d \log n/\delta})$. With probability $1 - \delta$, we have $\mathcal{M}' \subseteq \mathcal{M}_{O(\sqrt{d \log n/\delta})}$. By Claim C.1.6, applying Lemma C.1.5 to this set of distributions yields that we will select, with probability $1 - \delta$, a μ' so that $\|\mu - \mu'\|_2 \leq O(\varepsilon)$. By Claim C.1.6, this requires

$$O\left(\frac{\log |\mathcal{M}_{O(\sqrt{d \log n/\delta})}|}{\varepsilon^2}\right) = O\left(\frac{\log \binom{d}{k} + k \log(d/\varepsilon) + \log 1/\delta}{\varepsilon^2}\right) ,$$

samples, which simplifies to the desired bound, as claimed.

C.1.2 Proof of Upper Bound in Fact C.1.2

Our detection algorithm is given as follows. We let $\mathcal{N}$ be an $O(1)$-net over all k-sparse unit vectors, and we apply Lemma C.1.5 to the set $\{\mathcal{N}(0, I + \rho uu^\top)\}_{u \in \mathcal{N}} \cup \{\mathcal{N}(0, I)\}$. Clearly, we have:

Claim C.1.7. $|\mathcal{M}| = \binom{d}{k} 2^{O(k)}$.

By Fact C.1.4 and the guarantees of Lemma C.1.5, by an appropriate setting of parameters, if we have

$$n = O\left(\frac{\log |\mathcal{M}| + \log 1/\delta}{\varepsilon^2}\right) = O\left(\frac{k + \log \binom{d}{k} + \log 1/\delta}{\varepsilon^2}\right)$$

samples, then with probability $1 - \delta$ we will output $\mathcal{N}(0, I)$ if and only if the true model is $\mathcal{N}(0, I)$. This proves the upper bound.

C.2 Omitted Details from Section 3.4

C.2.1 Writing non-robust algorithms as dual norm maximization

In this section we will briefly review well-known non-robust algorithms for sparse mean recovery and for sparse PCA, and write them using our language.

Thresholding Recall that in the (non-robust) sparse mean estimation problem, one is given samples $X_1, \ldots, X_n \sim \mathcal{N}(\mu, I)$ where μ is k-sparse. The goal is then to recover μ. It turns out the simple thresholding algorithm ThresholdMean given in Algorithm 32 suffices for recovery:

Algorithm 32 Thresholding for sparse mean estimation

```
function ThresholdMean($X_1, \ldots, X_n$)
    Let $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$
    Let $S$ be the set of $k$ coordinates of $\widehat{\mu}$ with largest magnitude
    Let $\widehat{\mu}'$ be defined to be $\widehat{\mu}'_i = \widehat{\mu}_i$ if $i \in S$, 0 otherwise
    return $\widehat{\mu}'$
```

The correctness of this algorithm follows from the following folklore result, whose proof we shall omit for conciseness:

Fact C.2.1 (c.f. [RH17]). *Fix $\varepsilon, \delta > 0$, and let $X_1, \ldots, X_n$ be samples from $\mathcal{N}(\mu, I)$, where μ is k-sparse and*

$$n = \Omega\left(\frac{\log \binom{d}{k} + \log 1/\delta}{\varepsilon^2}\right) .$$

Then, with probability $1 - \delta$*, if* $\widehat{\mu}'$ *is the output of* ThresholdMean*, we have* $\|\widehat{\mu}' - \widehat{\mu}\|_2 \leq \varepsilon$*.*

To write this in our language, observe that

$$\text{ThresholdSMean}(X_1, \ldots, X_n) = \|\widehat{\mu}\|^*_{\mathcal{U}_k} \cdot d_{\mathcal{U}_k}(\widehat{\mu}) \ ,$$

where $\widehat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$.

L_1 **relaxation** In various scenarios, including recovery of a spiked covariance, one may envision the need to take k-sparse eigenvalues a matrix A, that is, vectors which solve the following non-convex optimization problem:

$$\begin{aligned} &\max v^\top A v \\ &\quad \text{s.t. } \|v\|_2 = 1, \ \|v\|_0 \leq k \ . \end{aligned} \tag{C.1}$$

However, this problem is non-convex and cannot by solved efficiently. This motivates the following SDP relaxation of (C.1): First, one rewrites the problem as

$$\begin{aligned} &\max \operatorname{tr}(AX) \\ &\quad \text{s.t. } \operatorname{tr}(X) = 1, \ \|X\|_0 \leq k^2 \ , X \succeq 0 \ , \operatorname{rank}(X) = 1 \end{aligned} \tag{C.2}$$

where $\|X\|_0$ is the number of non-zeros of X. Observe that since X is rank 1 if we let $X = vv^\top$ these two problems are indeed equivalent. Then to form the SDP, one removes the rank constraint, and relaxes the ℓ_0 constraint to a ℓ_1 constraint:

$$\begin{aligned} &\max \operatorname{tr}(AX) \\ &\quad \text{s.t. } \operatorname{tr}(X) = 1, \ \|X\|_1 \leq k \ , X \succeq 0 \ . \end{aligned} \tag{C.3}$$

The work of [dEGJL07] shows that this indeed detects the presence of a spike (but at an information theoretically suboptimal rate).

Finally, by definition, for any PSD matrix A, if X is the solution to (C.3) with input A, we have $X = d_{\mathcal{X}_k}(A)$.

C.2.2 Numerical precision

In general, we cannot find closed form solutions for $d_{\mathcal{X}_k}(A)$ in finite time. However, it is well-known that we can find these to very high numerical precision in polynomial time. For instance, using the ellipsoid method, we can find an M' so that $\|M' - d_{\mathcal{X}_k}(A)\|_\infty \leq \varepsilon$ in time $\mathrm{poly}(d, \log 1/\varepsilon)$. It is readily verified that if we set $\varepsilon' = \mathrm{poly}(\varepsilon, 1/d)$ then the numerical precision of the answer will not effect any of the calculations we make further on. Thus for simplicity of exposition we will assume throughout the paper that given any A, we can find $d_{\mathcal{X}_k}(A)$ exactly in polynomial time.

C.3 Computational Barriers for sample optimal robust sparse mean estimation

We conjecture that the rate achieved by Theorem 3.5.1 is tight for computationally efficient algorithms (up to log factors). Intuitively, the major difficulty is that distinguishing between $\mathcal{N}(\mu_1, I)$ and $\mathcal{N}(\mu_2, I)$ given corrupted samples seems to inherently require second moment (or higher) information, for any $\mu_1, \mu_2 \in \mathbb{R}^d$. Certainly first moment information by itself is insufficient. In this sparse setting, this is very problematic, as this inherently asks for us to detect a large sparse eigenvector of the empirical covariance. This more or less reduces to the problem solved by (C.1). This in turn requires us to relax to the problem solved by SDPs for sparse PCA, for which we know $\Omega(k^2 \log d/\varepsilon^2)$ samples are necessary for non-trivial behavior to emerge. We leave resolving this gap as an interesting open problem.

Appendix D

Deferred Details from Chapter 4

D.1 Toolkit for sum of squares proofs

Fact D.1.1 (See Fact A.1 in [MSS16] for a proof)**.** *Let $x_1, \ldots, x_n, y_1, \ldots, y_n$ be indeterminates. Then*

$$\vdash_4 \left(\sum_{i \le n} x_i y_i\right)^2 \le \left(\sum_{i \le n} x_i^2\right)\left(\sum_{i \le n} y_i^2\right) .$$

Fact D.1.2. *Let x, y be n-length vectors of indeterminates. Then*

$$\vdash_2 \|x+y\|_2^2 \le 2\|x\|_2^2 + 2\|y\|_2^2 .$$

Proof. The sum of squares proof of Cauchy-Schwarz implies that $\|x\|_2^2 + \|y\|_2^2 - 2\langle x, y\rangle$ is a sum of squares. Now we just expand

$$\|x+y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 + 2\langle x, y\rangle \preceq 2(\|x\|_2^2 + \|y\|_2^2) .$$

□

Fact D.1.3. *Let $P(x) \in \mathbb{R}[x]_\ell$ be a homogeneous degree ℓ polynomial in indetermi-*

nates $x = x_1, \ldots, x_n$. Suppose that the coefficients of P are bounded in 2-norm:

$$\sum_{\alpha \subseteq [n]} \hat{P}(\alpha)^2 \leq C \; .$$

(Here $\hat{P}(\alpha)$ are scalars such that $P(x) = \sum_\alpha \hat{P}(\alpha) x^\alpha$.) Let $a, b \in \mathbb{N}$ be integers such that $a + b = \ell$. Then

$$\vdash_{\max(2a,2b)} P(x) \leq \sqrt{C}(\|x\|_2^{2a} + \|x\|_2^{2b}) \; .$$

Proof. Let M be a matrix whose rows and columns are indexed by multisets $S \subseteq [n]$ of sizes a and b. Thus M has four blocks: an (a,a) block, an (a,b) block, a (b,a) block, and a (b,b) block. In the (a,b) and (b,a) blocks, put matrices M_{ab}, M_{ba} such that $\langle x^{\otimes a}, M_{ab} x^{\otimes b} \rangle = \frac{1}{2} .P(x)$. In the (a,a) and (b,b) blocks, put $\sqrt{C} \cdot I$. Then, letting $z = (x^{\otimes a}, x^{\otimes b})$, we get $\langle z, Mz \rangle = \sqrt{C}(\|x\|_2^{2a} + \|x\|_2^{2b}) - P(x)$. Note that $\|M_{ab}\|_F \leq \sqrt{C}$ by hypothesis, so $M \succeq 0$, which completes the proof. □

Fact D.1.4. *Let $u = (u_1, \ldots, u_k)$ be a vector of indeterminantes. Let D be sub-Gaussian with variancy proxy 1. Let $t \geq 0$ be an integer. Then we have*

$$\vdash_{2t} \mathop{\mathbb{E}}_{X \sim D} \langle X, u \rangle^{2t} \leq (2t)! \cdot \|u\|_2^{2t}$$
$$\vdash_{2t} \mathop{\mathbb{E}}_{X \sim D} \langle X, u \rangle^{2t} \geq -(2t)! \cdot \|u\|_2^{2t} \; .$$

Proof. Expand the polynomial in question. We have

$$\mathop{\mathbb{E}}_{X \sim D} \langle X, u \rangle^{2t} = \mathop{\mathbb{E}}_{X \sim D} \sum_\beta u^\beta \, \mathbb{E}[X^\beta] \; .$$

Let β range over $[k]^{2t}$

$$\vdash_{2t} \sum_\beta u^{2\beta} \, \mathbb{E} \, X^{2\beta} \leq (2t)! \sum_{\beta \text{ even}} u^\beta \leq \|u\|_2^{2t} \; .$$

where we have used upper bounds on the Gaussian moments $\mathbb{E} \, X^{2\beta}$ and that every term is a square in u. □

Fact D.1.5 (SoS Cauchy-Schwarz (see Fact A.1 in [MSS16] for a proof))**.** *Let $x_1, \ldots, x_n, y_1, \ldots, y_n$ be indeterminates. Then*

$$\vdash_4 \left(\sum_{i \le n} x_i y_i\right)^2 \le \left(\sum_{i \le n} x_i^2\right)\left(\sum_{i \le n} y_i^2\right) .$$

Fact D.1.6 (SoS Hölder)**.** *Let $w_1, \ldots, w_n$ and $x_1, \ldots, x_n$ be indeterminates. Let $q \in \mathbb{N}$ be a power of 2. Then*

$$\{w_i^2 = w_i \, \forall i \in [n]\} \vdash_{O(q)} \left(\sum_{i \le n} w_i x_i\right)^q \le \left(\sum_{i \le n} w_i\right)^{q-1} \cdot \left(\sum_{i \le n} x_i^q\right)$$

and

$$\{w_i^2 = w_i \, \forall i \in [n]\} \vdash_{O(q)} \left(\sum_{i \le n} w_i x_i\right)^q \le \left(\sum_{i \le n} w_i\right)^{q-1} \cdot \left(\sum_{i \le n} w_i \cdot x_i^q\right) .$$

Proof. We will only prove the first inequality. The second inequality follows since $w_i^2 = w_i \vdash_2 w_i x_i = w_i \cdot (w_i x_i)$, applying the first inequality, and observing that $w_i^2 = w_i \vdash_q w_i^q = w_i$.

Applying Cauchy-Schwarz (Fact D.1.1) and the axioms, we obtain to start that for any even number t,

$$\begin{aligned}\{w_i^2 = w_i \, \forall i \in [n]\} \vdash_{O(t)} \left[\left(\sum_{i \le n} w_i x_i\right)^2\right]^{t/2} &= \left[\left(\sum_{i \le n} w_i^2 x_i\right)^2\right]^{t/2} \\ &\le \left[\left(\sum_{i \le n} w_i^2\right)\left(\sum_{i \le n} w_i^2 x_i^2\right)\right]^{t/2} = \left(\sum_{i \le n} w_i\right)^{t/2}\left(\sum_{i \le n} w_i x_i^2\right)^{t/2} .\end{aligned}$$

It follows by indution that

$$\{w_i^2 = w_i \, \forall i \in [n]\} \vdash_{O(t)} \left[\left(\sum_{i \le n} w_i x_i\right)\right]^q \le \left(\sum_{i \le n} w_i\right)^{q-2}\left(\sum_{i \le n} w_i x_i^{q/2}\right)^2 .$$

Applying Fact D.1.1 one more time to get $\left(\sum_{i \le n} w_i x_i^{q/2}\right) \le \left(\sum_{i \le n} w_i^2\right)\left(\sum_{i \le n} x_i^q\right)$ and then the axioms $w_i^2 = w_i$ completes the proof. □

D.1.1 Examples of explicitly bounded distributions

In this section, we show that many natural high dimensional distributions are explicitly bounded. Recall that if a univariate distribution X *sub-Gaussian* (with variancy proxy σ) with mean μ then we have the following bound on its even centered moments for $t \ge 4$:

$$\mathbb{E}[(X-\mu)^t] \le \sigma^t \left(\frac{t}{2}\right)^{t/2} ,$$

if t is even.

More generally, we will say a univariate distribution is t-bounded with mean μ and variance proxy σ if the following general condition holds for all even $4 \le s \le t$:

$$\mathbb{E}[(X-\mu)^s] \le \sigma^s \left(\frac{s}{2}\right)^{s/2} .$$

The factor of 1/2 in this expression is not important and can be ignored upon first reading.

Our main result in this section is that any rotation of products of independent t-bounded distributions with variance proxy 1/2 is t-explicitly bounded with variance proxy 1:

Lemma D.1.7. *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d$ so that $\mathcal{D}$ is a rotation of a product distribution $\mathcal{D}'$ where each coordinate of $\mathcal{D}$ is a t-bounded univariate distribution with variance proxy $1/2$. Then $\mathcal{D}$ is t-explicitly bounded (with variance proxy 1).*

Proof. Since the definition of explicitly bounded is clearly rotation invariant, it suffices to show that $\mathcal{D}'$ is t-explicitly bounded. For any vector of indeterminants u, and for any $4 \leq s \leq t$ even, we have

$$\begin{aligned}\vdash_s \mathop{\mathbb{E}}_{X \sim \mathcal{D}'} \langle X - \mu, u \rangle^s &= \mathop{\mathbb{E}}_{X \sim \mathcal{D}'} \langle X - \mathop{\mathbb{E}}_{X' \sim \mathcal{D}'} X', u \rangle^s \\ &= \mathop{\mathbb{E}}_{X \sim \mathcal{D}'} \left(\mathop{\mathbb{E}}_{X'} \langle X - X', u \rangle \right)^s \\ &\leq \mathop{\mathbb{E}}_{X, X' \sim \mathcal{D}'} \langle X - X', u \rangle^s \ ,\end{aligned}$$

where X' is an independent copy of X, and the last line follows from SoS Cauchy-Schwarz. We then expand the resulting polynomial in the monomial basis:

$$\begin{aligned}\mathop{\mathbb{E}}_{X, X' \sim \mathcal{D}'} \langle X - X', u \rangle^s &= \sum_\alpha u^\alpha \mathop{\mathbb{E}}_{X, X'} (X - X')^\alpha \\ &= \sum_{\alpha \text{ even}} u^\alpha \mathop{\mathbb{E}}_{X, X'} (X - X')^\alpha \ ,\end{aligned}$$

since all α with odd monomials disappear since $X - X'$ is a symmetric product distribution. By t-boundedness, all remaining coefficients are at most s^{cs}, from which we deduce

$$\vdash_s \mathop{\mathbb{E}}_{X, X' \sim \mathcal{D}'} \langle X - X', u \rangle^s \leq s^{s/2} \sum_{\alpha \text{ even}} u^\alpha = s^{s/2} \|u\|_2^s \ ,$$

which proves that $\mathcal{D}'$ is t-explicitly bounded, as desired. □

As a corollary observe this trivially implies that all Guassians $\mathcal{N}(\mu, \Sigma)$ with $\Sigma \preceq I$ are t-explicitly bounded for all t.

We note that our results are tolerant to constant changes in the variancy proxy (just by scaling down). In particular, this implies that our results immediately apply for all rotations of products of t-bounded distributions with a loss of at most 2.

D.2 Sum of squares proofs for matrix positivity – omitted proofs

Lemma D.2.1 (Soundness)**.** *Suppose $\tilde{\mathbb{E}}$ is a degree-2d pseudodistribution which satisfies constraints $\{M_1 \succeq 0, \ldots, M_m \succeq 0\}$, and*

$$\{M_1 \succeq 0, \ldots, M_m \succeq 0\} \vdash_{2d} M \succeq 0 .$$

Then $\tilde{\mathbb{E}}$ satisfies $\{M_1 \succeq 0, \ldots, M_m \succeq 0, M \succeq 0\}$.

Proof. By hypothesis, there are r_S^j and B such that

$$M = B^\top \left[\sum_{S \subseteq [m]} \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \right] B .$$

Now, let $T \subseteq [m]$ and p be a polynomial. Let $M' = \otimes_{i \in T} M_i$. Suppose that $\deg(p^2 \cdot M \otimes M') \leq 2d$. Using the hypothesis on M, we obtain

$$\begin{aligned} p^2 \cdot M \otimes M' &= p^2 \cdot B^\top \left[\sum_{S \subseteq [m]} \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \right] B \otimes M' \\ &= (B \otimes I)^\top \left[p^2 \cdot \left[\sum_{S \subseteq [m]} \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \right] \otimes M' \right] (B \otimes I) . \end{aligned}$$

Applying $\tilde{\mathbb{E}}$ to the above, note that by hypothesis,

$$\tilde{\mathbb{E}} \left[p^2 \cdot \left[\sum_{S \subseteq [m]} \left(\sum_j (r_S^j(x))(r_S^j(x))^\top \right) \otimes [\otimes_{i \in S} M_i(x)] \right] \otimes M' \right] \succeq 0 .$$

The lemma follows by linearity. □

Lemma D.2.2. *Let $f(x)$ be a degree-ℓ s-vector-valued polynomial in indeterminates x. Let $M(x)$ be a $s \times s$ matrix-valued polynomial of degree ℓ'. Then*

$$\{M \succeq 0\} \vdash_{\ell \ell'} \langle f(x), M(x) f(x) \rangle \geq 0 .$$

Proof. Let $u \in \mathbb{R}^{s \otimes s}$ have entries $u_{ij} = 1$ if $i = j$ and otherwise $u_{ij} = 0$. Then $\langle f(x), M(x) f(x) \rangle = u^\top (M(x) \otimes f(x) f(x)^\top) u$. □

D.3 Omitted Proofs from Section 4.6

D.3.1 Proof of Lemma 4.6.4

We will show that each event (E1)–(E4) holds with probability at least $1 - d^{-8}$. Clearly for d sufficiently large this implies the desired guarantee. That (E1) and (E2) occur with probability $1 - d^{-8}$ follow from Lemmas 4.6.2 and 4.6.3, respectively. It now suffices to show (E3) and (E4) holds with high probability. Indeed, that (E4) holds with probability $1 - d^{-8}$ follows trivially from the same proof of Lemma 4.4.1 (it is in fact a simpler version of this fact).

Finally, we show that (E3) holds.

By basic concentration arguments (see e.g. [Ver10]), we know that by our choice of n, with probability $1 - d^{-8}$ we have that

$$\left\| \frac{1}{n} \sum_{i \in [n]} X_i - \mu^* \right\|_2 \leq \varepsilon \; . \tag{D.1}$$

Condition on the event that this and (E4) simultaneously hold. Recall that Y_i for $i = 1, \ldots, n$ are defined so that Y_i are iid and $Y_i = X_i$ for $i \in S_{\text{good}}$. By the triangle inequality, we have

$$\begin{aligned}\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i - \mu^* \right\|_2 &\leq \frac{n}{|S_{\text{good}}|} \left\| \frac{1}{n} \sum_{i \in [n]} Y_i - \mu^* \right\|_2 + \frac{|S_{\text{bad}}|}{|S_{\text{good}}|} \left\| \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} Y_i - \mu^* \right\|_2 \\ &\overset{(a)}{\leq} \frac{\varepsilon}{1 - \varepsilon} + \frac{|S_{\text{bad}}|}{|S_{\text{good}}|} \left\| \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} Y_i - \mu^* \right\|_2 \; , \end{aligned} \tag{D.2}$$

where (a) follows from (D.1).

We now bound the second term in the RHS. For any unit vector $u \in \mathbb{R}^d$, by

Hölder's inequality,

$$
\begin{aligned}
&\left\langle \sum_{i \in S_{\text{bad}}} (Y_i - \mu^*), u \right\rangle^t \le |S_{\text{bad}}|^{t-1} \sum_{i \in S_{\text{bad}}} \langle (Y_i - \mu^*), u \rangle^t \\
&\le |S_{\text{bad}}|^{t-1} \sum_{i \in [n]} \langle (Y_i - \mu^*), u \rangle^t \\
&\quad = |S_{\text{bad}}|^{t-1} \left[u^{\otimes t/2} \right]^\top \sum_{i \in [n]} \left[(Y_i - \mu^*)^{\otimes t/2} \right] \left[(Y_i - \mu^*)^{\otimes t/2} \right]^\top \left[u^{\otimes t/2} \right] \\
&\quad \overset{(a)}{\le} |S_{\text{bad}}|^{t-1} \cdot n \cdot \left[u^{\otimes t/2} \right]^\top \left(\mathop{\mathbb{E}}_{Y \sim D} \left[(Y - \mu^*)^{\otimes t/2} \right] \left[(Y - \mu^*)^{\otimes t/2} \right]^\top + \delta \cdot I \right) \left[(Y - \mu^*)^{\otimes t/2} \right] \\
&\quad = |S_{\text{bad}}|^{t-1} \cdot n \cdot \left(\mathop{\mathbb{E}}_{Y \sim D} \langle Y - \mu^*, u \rangle^t + \delta \right) \\
&\quad \le |S_{\text{bad}}|^{t-1} \cdot n \cdot (t^{t/2} + \delta) \\
&\quad \overset{(b)}{\le} 2 |S_{\text{bad}}|^{t-1} \cdot n \cdot t^{t/2} \; ,
\end{aligned}
$$

where (a) follows from (E4), and (b) follows since $\delta \ll t^t$. Hence

$$
\left\| \sum_{i \in S_{\text{bad}}} (Y_i - \mu^*) \right\|_2 = \max_{\|u\|_2 = 1} \left\langle \sum_{i \in S_{\text{bad}}} (Y_i - \mu^*), u \right\rangle \le O(|S_{\text{bad}}|^{1-1/t} \cdot n^{1/t} \cdot t^{1/2})
$$

Taking the t-th root on both sides and combining it with (D.2) yields

$$
\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i - \mu^* \right\|_2 \le \frac{\varepsilon}{1 - \varepsilon} + \frac{\varepsilon}{1 - \varepsilon} (n / |S_{\text{bad}}|)^{-1/t} \cdot t^{1/2} = O(\varepsilon^{1-1/t} \cdot t^{1/2}) \; ,
$$

as claimed.

D.4 Mixture models with nonuniform weights

In this section we describe at a high level how to adapt the algorithm given in Section 4.5 to handle non-uniform weights. We assume the mixture components now have mixture weights $\eta \le \lambda_1 \le \ldots \le \lambda_k \le 1$ where $\sum \lambda_i = 1$, where $\eta > 0$ is some fixed constant. We still assume that all pairs of means satisfy $\|\mu_i - \mu_j\|_2 \ge k^\gamma$ for all $i \neq j$. In this section we describe an algorithm LearnNonUniformMixtureModel, and we sketch a proof of the following theorem concerning its correctness:

Theorem D.4.1. *Let $\eta, \gamma > 0$ be fixed. Let $\mathcal{D}$ be a non-uniform mixture of k distributions $\mathcal{D}_1, \ldots, \mathcal{D}_k$ in $\mathbb{R}^d$, where each $\mathcal{D}_j$ is a $O(1/\gamma)$-explicitly bounded distribution with mean μ_j, and we have $\|\mu_i - \mu_j\|_2 \geq k^\gamma$. Furthermore assume that the smallest mixing weight of any component is at least η. Then, given $X_1, \ldots, X_n$ iid samples from $\mathcal{D}$ where $n \geq \frac{1}{\eta}(dk)^{O(1/\gamma)}$, $\textsc{LearnNonUniformMixtureModel}$ runs in $O(n^{1/t})$ time and outputs estimates $\widehat{\mu}_1, \ldots, \widehat{\mu}_m$ so that there is some permutation $\pi : [m] \to [m]$ so that $\|\widehat{\mu}_i - \mu_{\pi(i)}\|_2 \leq k^{-10}$ with probability at least $1 - k^{-5}$.*

Our modified algorithm is as follows: take n samples $X_1, \ldots, X_n$ where n is as in Theorem D.4.1. Then, do single-linkage clustering as before, and work on each cluster separately, so that we may assume without loss of generality that all means have pairwise ℓ_2 distance at most $O(\mathrm{poly}(d, k))$.

Within each cluster, we do the following. For $\alpha' = 1, 1 - \xi, 1 - 2\xi, \ldots, \eta$ for $\xi = \mathrm{poly}(\eta/k)$, iteratively form $\widehat{\mathcal{A}}$ with $\alpha = \alpha'$, $t = O\left(\frac{1}{\gamma}\right)$, and $\tau, \delta = k^{-10}$. Attempt to find a pseudo-expectation $\tilde{\mathbb{E}}$ that satisfies $\widehat{\mathcal{A}}$ with these parameters with minimal $\|\tilde{\mathbb{E}}\, ww^\top\|_F$. If none exists, then retry with the next α'. Otherwise, a rounding algorithm on $\tilde{\mathbb{E}}\, ww^\top$ to extract clusters. Remove these points from the dataset, and then continue with the next α'.

However, the rounding algorithm we require here is somewhat more involved than the naive rounding algorithm used previously for learning mixture models. In particular, we no longer know exactly the Frobenius norm of the optimal solution: we cannot give tight upper and lower bounds. This is because components with mixing weights which are just below the threshold α' may or may not contribute to the optimal solution that the SDP finds. Instead, we develop a more involved rounding algorithm $\textsc{RoundSecondMomentsNonuniform}$, which we describe below.

Our invariant is that every time we have a feasible solution to the SDP, we remove at least one cluster (we make this more formal below). Repeatedly run the SDP with this α' until we no longer get a feasible solution, and then repeat with a slightly smaller α'. After the loop terminates, output the empirical mean of every cluster.

The formal specification of this algorithm is given in Algorithm 33.

Algorithm 33 Mixture Model Learning

```
function LearnNonuniformMixtureMeans($t, \eta, X_1, \ldots, X_n$)
    Let $\xi \leftarrow \eta^2/(dk)^{-100}$
    Let $\mathcal{C} \leftarrow \{\}$, the empty set of clusters
    Let $\mathcal{X} \leftarrow \{X_1, \ldots, X_n\}$
    Perform naive clustering on $\mathcal{X}$ to obtain $\mathcal{X}_1, \ldots, \mathcal{X}_\ell$.
    for each $\mathcal{X}_r$ do
        Let $\alpha' \leftarrow 1$
        while $\alpha' \geq \eta - k^{-8}$ do
            By semidefinite programming (see Lemma 4.4.1, item 2), find a pseudo-expectation of degree $t = O(\frac{1}{\gamma})$ which satisfies the structured subset polynomials from Lemma 4.4.1, with $\alpha = \alpha' n$, and $\delta, \tau = k^{-8}$ with data points as in $\mathcal{X}$.
            while the SDP is feasible do
                Let $\tilde{\mathbb{E}}$ be the pseudoexpectation returned
                Let $M \leftarrow \tilde{\mathbb{E}} ww^\top$.
                Run the algorithm RoundSecondMomentsNonuniform on $M$ to obtain a cluster $C$.
                Let $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
                Remove all points in $C$ from $\mathcal{X}_r$
            Let $\alpha' \leftarrow \alpha' - \xi$
    return The empirical mean of every cluster in $\mathcal{C}$
```

For $j = 1, \ldots, k$ let S_j be the set of indices of points in $X_1, \ldots, X_n$ which were drawn from $\mathcal{D}_j$, and let $a_j \in \mathbb{R}^n$ be the indicator vectors for these sets as before. Our key invariant is the following: for every α' such that the SDP returns a feasible solution, we must have $|\alpha' - \lambda_j| \leq O(\xi)$ for some j, and moreover, for every j so that $\lambda_j \geq \alpha' + O(\xi)$, there must be exactly one cluster C_ℓ output by the algorithm at this point so that $|C_\ell \triangle S_j| \leq k^{-10} \operatorname{poly}(\eta) \cdot \cdot n$. Moreover, every cluster output so far must be of this form. For any α', we say that the algorithm up to α' is *well-behaved* if it satisfies this invariant for the loops in the algorithm for α'' for $\alpha'' > \alpha'$.

It is not hard to show, via arguments exactly as in Section 4.6 and 4.7 that the remaining fraction of points from these components which we have not removed as well as the small fraction of points we have removed from good components do not affect the calculations, and so we will assume for simplicity in the rest of this discussion that we have removed all samples from components j with $\lambda_j \geq \alpha' + O(\xi)$.

D.4.1 Sketch of proof of correctness of Algorithm 33

Here we outline the proof of correctness of Algorithm 33. The proof follows very similar ideas as the proof of correctness of Algorithm 11, and so for conciseness we omit many of the details. As before, for simplicity assume that the naive clustering returns only one cluster, as otherwise we can work on each cluster separately, so that for all i, we have $\|\mu_i\|_2 \leq O(\mathrm{poly}(d, k))$ after centering.

We now show why this invariant holds. Clearly this holds at the beginning of the algorithm. We show that if it holds at any step, it must also hold at the next time at which the SDP is feasible. Fix such an α'. By assumption, we have removed almost all points from components j with $\lambda_j \geq \alpha' + k^{-8}$, and have only removed a very small fraction of points not from these components.

By basic concentration, we have $|\lambda_j n - |S_j|| \leq o(n)$ for all j except with negligble probability, and so for the rest of the section, for simplicity, we will slightly cheat and assume that $\lambda_j n = |S_j|$. It is not hard to show that this also does not effect any calculations.

The main observation is that for any choice of α', by essentially same logic as in Section 4.5, we still have the following bound for all $i \neq j$ for an α' well-behaved run:

$$\widehat{\mathcal{A}} \vdash_{O(t)} \langle a_i, w\rangle \langle a_j, w\rangle \leq \frac{\eta n^2 t^{O(t)}}{k^{2t\gamma}} = O(\eta \xi^2) \cdot (\alpha')^2 n^2 \ , \tag{D.3}$$

for $\widehat{\mathcal{A}}$ instantiated with $\alpha = \alpha'$, where the last line follows by our choice of t sufficiently large.

We now show this implies:

Lemma D.4.2. *With parameters as above, for any α' well-behaved run, we have $\widehat{\mathcal{A}} \vdash_{O(t)} \langle a_i, w\rangle \leq O(\xi^2) \cdot \alpha' n$ for any j so that $\lambda_j n \leq (\alpha' - O(\xi^4))n$.*

Proof. We have

$$\widehat{\mathcal{A}} \vdash_t \sum_{j' \neq j} \langle a_i, w\rangle = \alpha' n - \langle a_j, w\rangle \geq \Omega(\xi^2) n \ ,$$

and hence

$$\begin{aligned}\widehat{\mathcal{A}} \vdash_{O(t)} \Omega(\xi^2) n \langle a_i, w\rangle &\le \langle a_i, w\rangle \sum_{j\neq i} \langle a_j, w\rangle \\ &\le \frac{1}{\eta} O(\eta\xi^4) \cdot (\alpha')^2 \cdot n^2 \ ,\end{aligned}$$

from which we deduce $\widehat{\mathcal{A}} \vdash_{O(t)} \langle a_i, w\rangle \le O(\xi^2) \cdot \alpha' n$. □

We now show that under these conditions, there is an algorithm to remove a cluster:

D.4.2 Rounding Well-behaved runs

Lemma D.4.3. *Let α', η, γ, t be as in Theorem D.4.1. Suppose that $\widehat{\mathcal{A}}$ is satisfiable with this set of parameters, that the algorithm has been α' well-behaved, and* (D.3) *holds. Then, there is an algorithm* RoundSecondMomentsNonuniform *which given $\tilde{\mathbb{E}}$ outputs a cluster C so that $|C \triangle S_j| \le (\eta/dk)^{O(1)} n$ with probability $1 - (\eta/dk)^{O(1)}$.*

Formally, let $v_i \in \mathbb{R}^n$ be so that for all i, j, we have $\langle v_i, v_j\rangle = \tilde{\mathbb{E}}\, w_i w_j$. Such v_i exist because $\tilde{\mathbb{E}}\, ww^\top$ is PSD, and can be found efficiently via spectral methods. For any cluster j, let V_j denote the set of vectors v_i for $i \in S_j$.

Our algorithm will proceed as follows: choose a random v_i with $\|v_i\|_2^2 \ge \alpha'/100$, and simply output as the cluster the set of ℓ so that $\|v_i - v_\ell\|_2 \le O(\sqrt{d}\xi)$.

We now turn to correctness of this algorithm. Define T to be the set of clusters j with $|\lambda_j - \alpha'| \le O(\xi^4)$. We first show:

Lemma D.4.4. *Assume that* (D.3) *holds. Then*

$$\sum_{\ell \in T} \sum_{i,j \in S_\ell} \|v_i - v_j\|_2^2 \le O(d^2\xi^2)(\alpha')^2 n^2 \ .$$

Proof. Observe that

$$\begin{aligned}\sum_{\ell\in T}\sum_{i,j\in S_\ell}\|v_i-v_j\|_2^2 &= \sum_{\ell\in T}\sum_{i,j\in S_\ell}\|v_i\|_2^2+\|v_j\|_2^2-2\langle v_i,v_j\rangle \\ &= \sum_{\ell\in T}\left(2|S_\ell|\sum_{i\in S_\ell}\|v_i\|_2^2-2\sum_{i,j\in S_\ell}\langle v_i,v_j\rangle\right)\ .\end{aligned}$$

By assumption, we have

$$\sum_{\ell\in T}\sum_{i\in S_\ell}|S_\ell|\|v_\ell\|_2^2 \;\;= (\alpha'\pm O(\xi^4))n\textstyle\sum_{\ell\in T}\|v_\ell\|_2^2 \;\;= (\alpha'\pm O(\xi^4))n\cdot\tilde{\mathbb{E}}\left(\displaystyle\sum_{\ell\in T}\sum_{i\in S_\ell}w_i^2\right)\ .$$

Since by Lemma D.4.2 we have $\tilde{\mathbb{E}}[\sum_{\ell\notin T}\sum_{i\in S_\ell}w_i^2]\le dO(\xi^2)\alpha n$, we conclude that

$$\alpha n\ge\tilde{\mathbb{E}}\left(\sum_{\ell\in T}\sum_{i\in S_\ell}w_i^2\right)\ge(1-dO(\xi^2))\alpha' n\ .$$

All of this allows us to conclude

$$\sum_{\ell\in T}\sum_{i\in S_\ell}|S_\ell|\|v_\ell\|_2^2=(1\pm O(d\xi^2))(\alpha')^2n^2\ .$$

On the other hand, we have

$$\sum_{\ell\in T}\sum_{i,j\in S_\ell}\langle v_i,v_j\rangle=\sum_{\ell\in T}\tilde{\mathbb{E}}\langle a_\ell,w\rangle^2\ ,$$

but we have

$$\begin{aligned}(\alpha')^2n^2 &= \tilde{\mathbb{E}}\left(\sum_\ell\langle a_\ell,w\rangle\right)^2 \\ &= \sum_{\ell\neq j}\tilde{\mathbb{E}}[\langle a_\ell,w\rangle\langle a_j,w\rangle]+\sum_{\ell\notin T}\langle a_\ell,w\rangle^2+\sum_{\ell\in T}\langle a_\ell,w\rangle^2\ .\end{aligned}$$

The first term is at most $O(d^2\eta\xi^2)(\alpha')^2n^2$ by (D.3) and the second term is at most

$dO(\xi^2)\alpha' n$ by Lemma D.4.2, so overall we have that

$$\sum_{\ell \in T} \tilde{\mathbb{E}} \langle a_\ell, w \rangle^2 \quad = (1 \pm O(d^2 \xi^2))(\alpha')^2 n^2 \ .$$

Hence putting it all together we have

$$\sum_{\ell \in T} \sum_{i,j \in S_\ell} \|v_i - v_j\|_2^2 \quad = O(d^2 \xi^2)(\alpha')^2 n^2 \ ,$$

as claimed. □

As a simple consequence of this we have:

Lemma D.4.5. *Assume that* (D.3) *holds. For all $\ell \in T$, there exists a ball B of radius $O(\sqrt{d\xi})$ so that $|V_\ell \triangle B| \leq O(d\xi)\alpha' n$.*

Proof. Suppose not, that is, for all B with radius $O(d\xi)$, we have $|S_\ell \triangle B| \leq \Omega(d\xi)\alpha' n$. Consider the ball of radius $O(\sqrt{m\xi})$ centered at each v_i for $i \in S_\ell$. By assumption there are $\Omega(d\xi)\alpha' n$ vectors outside the ball, that is, with distance at least $\Omega(\sqrt{d\xi})$ from v_i. Then

$$\sum_{i,j \in S_\ell} \|v_i - v_j\|_2^2 \geq n \cdot \Omega(d\xi)\Omega(d\xi)\alpha n \geq \Omega(d^2\xi^2)\alpha' n \ ,$$

which contradicts the previous lemma. □

Associate to each cluster $\ell \in T$ a ball B_ℓ so that $|V_\ell \triangle B| \leq \Omega(d\xi)\alpha' n$. Let ϕ_ℓ denote the center of B_ℓ. We now show that if we have two j, ℓ so that either $\|\phi_j\|_2$ or $\|\phi_\ell\|_2$ is large, then B_ℓ and B_j must be disjoint. Formally:

Lemma D.4.6. *Assume that* (D.3) *holds. Let $j, \ell \in T$ so that $\|\phi_j\|_2^2 + \|\phi_\ell\|_2^2 \geq \Omega(\alpha')$. Then $B_j \cap B_\ell = \varnothing$.*

Proof. We have

$$\begin{aligned}
\sum_{i \in B_j, k \in B_\ell} \|v_i - v_k\|_2^2 &= \sum_{i \in B_j, k \in B_\ell} \|v_i\|_2^2 + \|v_k\|_2^2 - 2\langle v_i, v_k \rangle \\
&= |B_\ell| \sum_{i \in B_j} \|v_i\|_2^2 + |B_j| \sum_{k \in B_\ell} \|v_k\|_2^2 - 2 \sum_{i \in B_j, k \in B_\ell} \tilde{\mathbb{E}}\, w_i w_k \\
&\geq (\alpha' - O(\xi^4))n \left(\sum_{i \in B_j} \|v_i\|_2^2 + |B_j| \sum_{k \in B_\ell} \|v_k\|_2^2 \right) - 2\, \tilde{\mathbb{E}} \langle a_j, w \rangle \langle a_\ell, w \rangle \\
&\geq (\alpha' - O(\xi^4))n \left(\sum_{i \in B_j} \|v_i\|_2^2 + \sum_{i \in B_k} \|v_k\|_2^2 \right) - O(\eta \xi^2)(\alpha')^2 n^2 \;.
\end{aligned}$$

Observe that

$$\begin{aligned}
\sum_{i \in B_j} \|v_i\|_2^2 &= \sum_{i \in B_j, v_i \in B_j} \|v_i\|_2^2 + \sum_{\in B_j, v_i \notin B_j} \|v_i\|_2^2 \\
&\geq (1 - O(d\xi))\alpha' n \left(\|\phi_0\|_2^2 - d\xi \right) + O(d\xi)\alpha' n \\
&\geq \alpha' n \|\phi_0\|_2^2 - O(m\xi)\alpha' n \;.
\end{aligned}$$

since generically $\|v_i\|_2^2 = \tilde{\mathbb{E}}\, w_i^2 \leq 1$. Symmetrically we have $\sum_{k \in B_\ell} \|v_k\|_2^2 \geq (\|\phi_1\|_2^2 - O(d\xi))\alpha' n$. Hence we have

$$\sum_{i \in B_j, k \in B_\ell} \|v_i - v_k\|_2^2 \;\; \geq (\|\phi_1\|_2^2 + \|\phi_2\|_2^2 - O(m\xi))(\alpha')^2 n^2 \geq \Omega(\alpha')^2 \cdot (\alpha')^2 n^2 \;.$$

Now suppose that $B_j \cap B_\ell \neq \varnothing$. This implies that for all except for a $O(d\xi)(\alpha')^2 n^2$ set of pairs i, j (i.e. those containing $v_i \notin B_j$ or $v_j \notin B_\ell$), the pairwise squared distance is at most $O(d\xi)$. Since the pairwise distance between any two points is at most 2, this is a clear contradiction. □

Finally, we show that a random point with large norm will likely be within a B_ℓ.

Lemma D.4.7. *Let i be a uniformly random index over the set of indices so that $\|v_i\|_2^2 \geq \alpha'/100$. Then, with probability $1 - O(d\xi)$, $v_i \in B_\ell$ for some ℓ.*

Proof. Observe that since $\|v_i\|_2^2 \leq 1$ and $\sum \|v_i\|_2^2 = \alpha' n$ there are at least $(1 -$

$1/100)\alpha' n$ vectors with $\|v_i\|_2^2 \geq \alpha'/100$. We have

$$\sum_{\ell \notin T} \|v_i\|_2^2 = \sum_{\ell \notin T} \tilde{\mathbb{E}} \langle a_\ell, w \rangle \leq O(d\xi^2)\alpha' n \ ,$$

so by Markov's inequality the number of i with $i \in \cup_{\ell \notin T} S_\ell$ and $\|v_i\|_2^2 \geq \alpha'/100$ is at most $100 \cdot O(d\xi^2)n \ll O(m\xi)\alpha' n$. There are at most $O(d\xi)\alpha' n$ vectors v_i so that $v_i \in S_\ell$ for $\ell \in T$ and $v_i \notin B_\ell$, and so the probability that a vector with $\|v_i\|_2^2 \geq \alpha'/100$ is not of the desired form is at most $O(d\xi)$, as claimed. □

This completes the proof of Lemma D.4.3, since this says that if we choose i uniformly at random amongst all such $\|v_i\|_2^2 \geq \alpha/100$, then with probability $1 - O(d\xi)$, we have $v_i \in B_\ell$ for some B_ℓ with $\|\phi_\ell\|_2 = \Omega(\alpha')$, and hence if we look in a $O(\sqrt{d\xi})$ ball around it, it will contain all but a $O(d\xi)\alpha' n$ fraction of points from S_ℓ.

Appendix E

Deferred Proofs from Chapter 5

E.1 Proof of Lemma 5.3.2

In fact, we will prove a stronger statement, which clearly implies Lemma 5.3.2. Namely, we will show that it holds for general sub-gaussian distributions. This will in particular be important to show that our algorithm works for isotropic sub-gaussian distributions.

Lemma E.1.1. *Let $\varepsilon, \delta > 0$. Let G be a subgaussian distribution over $\mathbb{R}^d$ with mean μ and variance proxy 1. Let S be a set of n i.i.d. samples from G, where*

$$n = \Omega\left(\frac{d}{\varepsilon^2} \operatorname{poly}\log \frac{d}{\varepsilon\delta}\right) .$$

Then with probability $1-\delta$, S satisfies (5.7)-(5.10).

Proof. For (5.7), the probability that a coordinate of a sample after centering by μ is at least $\sqrt{2\log(nd/(3\delta)}$ is at most $\frac{\delta}{3dn}$ by Fact 1.4.1. By a union bound, the probability that all coordinates of all samples are smaller than $\sqrt{2\log(nd/(3\delta))}$ is at least $1-\tau/3$. In this case, $\|x\|_2 \le \sqrt{2d\log\frac{nd}{3\tau}} = O\left(\sqrt{d\log\frac{n}{\tau}}\right)$.

After translating by μ, we note that (5.8) follows immediately from Lemma 2.1.6

and (5.9) follows from Theorem 5.50 of [Ver10], as long as

$$n = \Omega\left(\frac{d + \log(1/\delta)}{\varepsilon^2}\right) ,$$

with probability at least $1 - \delta/3$. It remains to show that, conditioned on (5.7)-(5.9), (5.10) holds with probability at least $1 - \delta/3$.

To simplify some expressions, let $\rho := \varepsilon/(\log(d\log(d/\varepsilon\delta)))$ and $R = C\sqrt{d\log(|S|/\tau)}$ for some universal constant C sufficiently large. We need to show that for all unit vectors v and all $0 \leq T \leq R$ that

$$\left|\Pr_{X \in_u S}[|v \cdot (X - \mu)| > T] - \Pr_{X \sim G}[|v \cdot (X - \mu) > T \geq 0]\right| \leq \frac{\rho}{T^2} . \tag{E.1}$$

Firstly, we have that for all unit vectors v and $T > 0$

$$\left|\Pr_{X \in_u S}[|v \cdot (X - \mu)| > T] - \Pr_{X \sim G}[|v \cdot (X - \mu)| > T \geq 0]\right| \leq \frac{\rho}{10\ln(1/\rho)}$$

with probability at least $1 - \delta/6$. Since the VC-dimension of the set of all halfspaces is $d + 1$, this follows from the VC inequality [DL12], since we have more than $\Omega(d/(\rho/(10\log(1/\rho))^2)$ samples. We thus only need to consider the case when $T \geq \sqrt{10\ln(1/\rho)}$.

To handle this case, we show:

Lemma E.1.2. *For any fixed unit vector v and $T > \sqrt{10\ln(1/\rho)}$, except with probability $\exp(-N\rho/(6C))$, we have that*

$$\Pr_{X \in_u S}[|v \cdot (X - \mu)| > T] \leq \frac{\rho}{CT^2} ,$$

where $C = 8$.

Proof. Let E be the event that $|v \cdot (X - \mu)| > T$. Since G is sub-gaussian, Fact 1.4.1 yields that

$$\Pr_G[E] = \Pr_{Y \sim G}[|v \cdot (X - \mu)| > T] \leq \exp(-T^2/(2\nu)) .$$

Note that, thanks to our assumption on T, we have that $T \leq \exp(T^2/(4))/2C$, and therefore $T^2 \Pr_G[E] \leq \exp(-T^2/(4))/2C \leq \rho/2C$.

Consider $\mathbb{E}_S[\exp(t^2/(3) \cdot N \Pr_S[E])]$. Each individual sample X_i for $1 \leq i \leq N$, is an independent copy of $Y \sim G$, and hence:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S}\left[\exp\left(\frac{T^2}{3} \cdot n \Pr_S[E]\right)\right] &= \mathop{\mathbb{E}}_{S}\left[\exp\left(\frac{T^2}{3}\right) \cdot \sum_{i=1}^{n} 1_{X_i \in E})\right] \\
&= \prod_{i=1}^{n} \mathop{\mathbb{E}}_{X_i}\left[\exp\left(\frac{T^2}{3}\right) \cdot \sum_{i=1}^{n} 1_{X_i \in E})\right] \\
&= \left(\exp\left(\frac{T^2}{3}\right) \Pr_G[E] + 1\right)^n \\
&\overset{(a)}{\leq} \left(\exp\left(\frac{T^2}{6}\right) + 1\right)^n \\
&\overset{(b)}{\leq} (1 + \rho^{5/3})^n \\
&\overset{(c)}{\leq} \exp(n\rho^{5/3}) ,
\end{aligned}$$

where (a) follows from sub-gaussianity, (b) follows from our choice of T, and (c) comes from the fact that $1 + x \leq e^x$ for all x.

Hence, by Markov's inequality, we have

$$\begin{aligned}
\Pr\left[\Pr_S[E] \geq \frac{\rho}{CT^2}\right] &\leq \exp\left(N\rho^{5/3} - \frac{\rho n}{3C}\right) \\
&= \exp(n\rho(\rho^{2/3} - 1/(3C))) .
\end{aligned}$$

Thus, if δ is a sufficiently small constant and C is sufficiently large, this yields the desired bound. □

Now let $\mathcal{C}$ be a 1/2-cover in Euclidean distance for the set of unit vectors of size $2^{O(d)}$. By a union bound, for all $v' \in \mathcal{C}$ and T' a power of 2 between $\sqrt{4\ln(1/\delta)}$ and R, we have that

$$\Pr_{X \in_u S}[|v' \cdot (X - \mu)| > T'] \leq \frac{\rho}{8T^2}$$

except with probability

$$2^{O(d)}\log(R)\exp(-n\rho/6C) = \exp\left(O(d) + \log\log R - n\rho/6C\nu\right) \leq \delta/6 \ .$$

However, for any unit vector v and $\sqrt{4\ln(1/\rho)} \leq T \leq R$, there is a $v' \in \mathcal{C}$ and such a T' such that for all $x \in \mathbb{R}^d$, we have $|v \cdot (X-\mu)| \geq |v' \cdot (X-\mu)|/2$, and so $|v' \cdot (X-\mu)| > 2T'$ implies $|v' \cdot (X-\mu)| > T$.

Then, by a union bound, (E.1) holds simultaneously for all unit vectors v and all $0 \leq T \leq R$, with probability a least $1-\delta/3$. This completes the proof. □

E.2 Proof of Lemma 5.4.2

Proof of Lemma 5.4.2: Note that an even polynomial has no degree-1 terms. Thus, we may write $p(x) = \sum_i p_{i,i}x_i^2 + \sum_{i>j} p_{i,j}x_ix_j + p_o$. Taking $(P_2)_{i,i} = p_{i,i}$ and $(P_2')_{i,j} = (P_2')_{j,i} = \frac{1}{2}p_{i,j}$, for $i > j$, gives that $p(x) = x^T P_2' x + p_0$. Taking $P_2 = \Sigma^{1/2}P_2'\Sigma^{1/2}$, we have $p(x) = (\Sigma^{-1/2}x)^T P_2(\Sigma^{-1/2}x) + p_0$, for a $d \times d$ symmetric matrix P_2 and $p_0 \in \mathbb{R}$.

Let $P_2 = U^T\Lambda U$, where U is orthogonal and Λ is diagonal be an eigen-decomposition of the symmetric matrix P_2. Then, $p(x) = (U\Sigma^{-1/2}x)^T P_2(U\Sigma^{-1/2}x)$. Let $X \sim G$ and $Y = U\Sigma^{-1/2}X$. Then, $Y \sim \mathcal{N}(0,I)$ and $p(X) = \sum_i \lambda_i Y_i^2 + p_0$ for independent Gaussians Y_i. Thus, $p(X)$ follows a generalized χ^2-distribution.

Thus, we have

$$\mathbb{E}[p(X)] = \mathbb{E}\left[\sum_i \lambda_i Y_i^2 + p_0\right] = p_0 + \sum_i \lambda_i = p_0 + \mathrm{tr}(P_2) \ ,$$

and

$$\mathrm{Var}[p(X)] = \mathrm{Var}\left[\sum_i \lambda_i Y_i^2 + p_0\right] = \sum_i \lambda_i^2 = \|P_F\|_2 \ .$$

Lemma E.2.1 (cf. Lemma 1 from [LM00]). *Let $Z = \sum_i a_i Y_i^2$, where Y_i are independent random variables distributed as $\mathcal{N}(0,1)$. Let a be the vector with coordinates a_i. Then,*

$$\Pr(Z \geq 2\|a\|_2\sqrt{x} + 2\|a\|_\infty x) \leq \exp(-x) \ .$$

We thus have:

$$\Pr\left(\sum_i \lambda_i(Y_i^2-1) > 2\sqrt{(\sum_i \lambda_i^2)t} + 2(\max_i \lambda_i)t\right) \leq e^{-t} \ .$$

Noting that $\mathrm{tr}(P_2) = \sum_i \lambda_i$,$\sum_i \lambda_i^2 = \|P_2\|_F$ and $\max_i \lambda_i = \|P_2\|_2 \leq \|P_2\|$, for $\mu_p = \mathbb{E}[p(X)]$ we have:

$$\Pr(p(X) - \mu_p > 2\|P_2\|_F(\sqrt{t}+t)) \leq e^{-t} \ .$$

Noting that $2\sqrt{a} = 1 + a - (1-\sqrt{a})^2 \leq 1 + a$ for $a > 0$, we have

$$\Pr(p(X) - \mu_p > \|P_2\|_F(3t+1)) \leq e^{-t} \ .$$

Applying this for $-p(x)$ instead of $p(x)$ and putting these together, we get

$$\Pr(|p(X) - \mu_p| > \|P_2\|_F(3t+1)) \leq 2e^{-t} \ .$$

Substituting $t = T/3\|P_2\|_F - 1/3$, and $2\|P_2\|_F^2 = \mathrm{Var}_{X\sim G}(p(X))$ gives:

$$\Pr(|p(X) - \mathop{\mathbb{E}}_{X\sim G}[p(X)]| \geq T) \leq 2e^{1/3 - 2T/3\,\mathrm{Var}_{X\sim G}[p(X)]} \ .$$

The final property is a consequence of the following anti-concentration inequality:

Theorem E.2.2 ([CW01]). *Let $p : \mathbb{R}^d \to \mathbb{R}$ be a degree-d polynomial. Then, for $X \sim \mathcal{N}(0,I)$, we have*

$$\Pr(|p(X)| \leq \varepsilon\sqrt{\mathbb{E}[p(X)^2]} \leq O(d\varepsilon^{1/d}) \ .$$

This completes the proof. □

E.3 Proof of Lemma 5.4.3

Proof of Lemma 5.4.3: Firstly, we note that it suffices to prove this for the case $\Sigma = I$, since for $X \sim \mathcal{N}(0,\Sigma)$, $Y = \Sigma^{-1/2}X$ is distributed as $\mathcal{N}(0,I)$, and all the conditions transform to those for $G = \mathcal{N}(0, I)$ under this transformation.

Condition 5.18 follows by standard concentration bounds on $\|x\|_2^2$. Condition 5.19 follows by estimating the entry-wise error between $\mathrm{Cov}(S)$ and I. These two conditions hold by Lemma 5.3.2.

Condition 5.20 is slightly more involved. Let $\{p_i\}$ be an orthonormal basis for the set of even, degree-2, mean-0 polynomials with respect to G. Define the matrix $M_{i,j} = \mathbb{E}_{x \in_u S}[p_i(x)p_j(x)] - \delta_{i,j}$. This condition is equivalent to $\|M\|_2 = O(\varepsilon)$. Thus, it suffices to show that for every v with $\|v\|_2 = 1$ that $v^T M v = O(\varepsilon)$. It actually suffices to consider a cover of such v's. Note that this cover will be of size $2^{O(d^2)}$. For each v, let $p_v = \sum_i v_i p_i$. We need to show that $\mathrm{Var}(p_v(S)) = 1 + O(\varepsilon)$. We can show this happens with probability $1 - \tau 2^{-\Omega(d^2)}$, and thus it holds for all v in our cover by a union bound.

Condition 5.21 is substantially the most difficult of these conditions to prove. Naively, we would want to find a cover of all possible p and all possible T, and bound the probability that the desired condition fails. Unfortunately, the best a priori bound on $\Pr(|p(G)| > T)$ are on the order of $\exp(-T)$. As our cover would need to be of size 2^{d^2} or so, to make this work with $T = d$, we would require on the order of d^3 samples in order to make this argument work.

However, we will note that this argument is sufficient to cover the case of $T < 10 \log(1/\varepsilon) \log^2(d/\varepsilon)$.

Fortunately, most such polynomials p satisfy much better tail bounds. Note that any even, mean zero polynomial p can be written in the form $p(x) = x^T A x - \mathrm{tr}(A)$ for some matrix A. We call A the associated matrix to p. We note by the Hanson-Wright inequality that $\Pr_{X \sim G}(|p(X)| > T) = \exp(-\Omega(\min((T/\|A\|_F)^2, T/\|A\|_2)))$. Therefore, the tail bounds above are only as bad as described when A has a single large eigenvalue. To take advantage of this, we will need to break p into parts based on the size of its eigenvalues. We begin with a definition:

Definition E.3.1. Let $\mathcal{P}_k$ be the set of even, mean-0, degree-2 polynomials, such that the associated matrix A satisfies:

1. $\operatorname{rank}(A) \leq k$

2. $\|A\|_2 \leq 1/\sqrt{k}$.

Note that for $p \in \mathcal{P}_k$ that $|p(x)| \leq |x|^2/\sqrt{k} + \sqrt{k}$.

Importantly, any polynomial can be written in terms of these sets.

Lemma E.3.1. *Let p be an even, degree-2 polynomial with $\mathbb{E}_{X\sim G}[p(X)] = 0, \operatorname{Var}_{X\sim G}(p(X)) = 1$. Then if $t = \lfloor \log_2(d) \rfloor$, it is possible to write $p = 2(p_1 + p_2 + \ldots + p_{2^t} + p_d)$ where $p_k \in \mathcal{P}_k$.*

Proof. Let A be the associated matrix to p. Note that $\|A\|_F = \operatorname{Var} p = 1$. Let A_k be the matrix corresponding to the top k eigenvalues of A. We now let p_1 be the polynomial associated to $A_1/2$, p_2 be associated to $(A_2 - A_1)/2$, p_4 be associated to $(A_4 - A_2)/2$, and so on. It is clear that $p = 2(p_1 + p_2 + \ldots + p_{2^t} + p_d)$. It is also clear that the matrix associated to p_k has rank at most k. If the matrix associated to p_k had an eigenvalue more than $1/\sqrt{k}$, it would need to be the case that the $k/2^{nd}$ largest eigenvalue of A had size at least $2/\sqrt{k}$. This is impossible since the sum of the squares of the eigenvalues of A is at most 1.

This completes our proof. □

We will also need covers of each of these sets $\mathcal{P}_k$. We will assume that condition 5.18 holds, i.e., that $\|x\|_2 \leq \sqrt{R}$, where $R = O(d \log(d/\varepsilon\tau))$. Under this condition, $p(x)$ cannot be too large and this affects how small a variance polynomial we can ignore.

Lemma E.3.2. *For each k, there exists a set $\mathcal{C}_k \subset \mathcal{P}_k$ such that*

1. *For each $p \in \mathcal{P}_k$ there exists a $q \in \mathcal{C}_k$ such that $\operatorname{Var}_{X\sim G}(p(X) - q(X)) \leq 1/R^2 d^2$.*

2. *$|\mathcal{C}_k| = 2^{O(dk \log R)}$.*

Proof. We note that any such p is associated to a matrix A of the form $A = \sum_{i=1}^k \lambda_i v_i v_i^T$, for $\lambda_i \in [0, 1/\sqrt{k}]$ and v_i orthonormal. It suffices to let q correspond to the matrix $A' = \sum_{i=1}^k \mu_i w_i w_i^T$ for with $|\lambda_i - \mu_i| < 1/R^2 d^3$ and $|v_i - w_i| < 1/R^2 d^3$

for all i. It is easy to let μ_i and w_i range over covers of the interval and the sphere with appropriate errors. This gives a set of possible q's of size $2^{O(dk \log R)}$ as desired. Unfortunately, some of these q will not be in $\mathcal{P}_k$ as they will have eigenvalues that are too large. However, this is easily fixed by replacing each such q by the closest element of $\mathcal{P}_k$. This completes our proof. □

We next will show that these covers are sufficient to express any polynomial.

Lemma E.3.3. *Let $p \in \mathcal{P}_2(\Sigma)$. It is possible to write p as a sum of $O(\log(d))$ elements of some $\mathcal{C}_k$ plus another polynomial of variance at most $O(1/R^2)$.*

Proof. Combining the above two lemmata we have that any such p can be written as

$$p = (q_1 + p_1) + (q_2 + p_2) + \ldots (q_{2^t} + p_{2^t}) + (q_d + p_d) = q_1 + q_2 + \ldots + q^{2^t} + q^d + p' ,$$

where q_k above is in $\mathcal{C}_k$ and $\mathrm{Var}_{X \sim G}[p_k(X)] < 1/R^2 d^2$. Thus, $p' = p_1 + p_2 + \ldots + p_{2^t} + p_d$ has $\mathrm{Var}_{X \sim G}[p'(X)] \leq O(1/R^2)$. This completes the proof. □

The key observation now is that if $|p(x)| \geq T$ for $\|x\|_2 \leq \sqrt{d/\varepsilon}$, then writing $p = q_1 + q_2 + q_4 + \ldots + q_d + p'$ as above, it must be the case that $|q_k(x)| > (T-1)/(2\log(d))$ for some k. Therefore, to prove our main result, it suffices to show that, with high probability over the choice of S, for any $T \geq 10 \log(1/\varepsilon) \log^2(d/\varepsilon)$ and any $q \in \mathcal{C}_k$ for some k, that $\Pr_{x \in_u S}(|q(x)| > T/(2\log(d))) < \varepsilon/(2T^2 \log^2(T) \log(d))$. Equivalently, it suffices to show that for $T \geq 10 \log(1/\varepsilon) \log(d/\varepsilon)$ it holds $\Pr_{x \in_u S}(|q(x)| > T/(2\log(d))) < \varepsilon/(2T^2 \log^2(T) \log^2(d))$. Note that this holds automatically for $T > R$, as $p(x)$ cannot possibly be that large for $\|x\|_2 \leq \sqrt{R}$. Furthermore, note that losing a constant factor in the probability, it suffices to show this only for T a power of 2.

Therefore, it suffices to show for every $k \leq d$, every $q \in \mathcal{C}_k$ and every $R/\sqrt{k} \gg T \gg \log(1/\varepsilon) \log R$ that with probability at least $1 - \tau 2^{-\Omega(dk \log R)}$ over the choice of S we have that $\Pr_{x \in_u S}(|q(x)| > T) \ll \varepsilon/(T^2 \log^4(R))$. However, by the Hanson-Wright inequality, we have that

$$\Pr_{X \sim G}(|q(X)| > T) = \exp(-\Omega(\min(T^2, T\sqrt{k}))) < (\varepsilon/(T^2 \log^4 R))^2 .$$

Therefore, by Chernoff bounds, the probability that more than a $\varepsilon/(T^2 \log^4 R)$-fraction of the elements of S satisfy this property is at most

$$\begin{aligned}\exp(-\Omega(\min(T^2, T\sqrt{k}))|S|\varepsilon/(T^2 \log^4 R)) &= \exp(-\Omega(|S|\varepsilon/(\log^4 R) \min(1, \sqrt{k}/T))) \\ &\leq \exp(-\Omega(|S|k\varepsilon^2/R(\log^4 R))) \\ &\leq \exp(-\Omega(|S|k\varepsilon/d(\log(d/\varepsilon\tau))(\log^4(d/\log(1/\varepsilon\tau))))) \\ &\leq \tau \exp(-\Omega(dk \log(d/\varepsilon))) \ ,\end{aligned}$$

as desired.

This completes our proof.

□

Appendix F

Omitted Details from Chapter 6

F.1 Full description of the distributions for synthetic and semi-synthetic experiments

Here we formally describe the distributions we used in our experiments. In all settings, our goal was to find noise distributions so that noise points were not "obvious" outliers, in the sense that there is no obvious pointwise pruning process which could throw away the noise points, which still gave the algorithms we tested the most difficulty. We again remark that while other algorithms had varying performances depending on the noise distribution, it seemed that the performance of ours was more or less unaffected by it.

Distribution for the synthetic mean experiment Our uncorrupted points were generated by $\mathcal{N}(\mu, I)$, where μ is the all-ones vector. Our noise distribution is given as

$$N = \frac{1}{2}\Pi_1 + \frac{1}{2}\Pi_2 \ ,$$

where Π_1 is the product distribution over the hypercube where every coordinate is 0 or 1 with probability $1/2$, and Π_2 is a product distribution where the first coordinate is ether 0 or 12 with equal probability, the second coordinate is -2 or 0 with equal probability, and all remaining coordinates are zero.

Distribution for the synthetic covariance experiment For the isotropic synthetic covariance experiment, our uncorrupted points were generated by $\mathcal{N}(0, I)$, and the noise points were all zeros. For the skewed synthetic covariance experiment, our uncorrupted points were generated by $\mathcal{N}(0, I + 100 e_1 e_1^T)$, where e_1 is the first unit vector, and our noise points were generated as follows: we took a fixed random rotation of points of the form $Y_i \sim \Pi$, where Π is a product distribution whose first $d/2$ coordinates are each uniformly selected from $\{-0.5, 0, 0.5\}$, and whose next $d/2 - 1$ coordinates are each $0.8 \times A_i$, where for each coordinate i, A_i is an independent random integer between -2 and 2, and whose last coordinate is a uniformly random integer between $[-100, 100]$.

Setup for the semi-synthetic geographic experiment We took the 20 dimensional data from [NJB$^+$08], which was diagonalized, and randomly rotated it. This was to simulate the higher dimensional case, since the singular vectors that [NJB$^+$08] obtained did not seem to be sparse or analytically sparse. Our noise was distributed as Π, where Π is a product distribution whose first $d/2$ coordinates are each uniformly random integers between 0 and 2 and whose last $d/2$ coordinates are each uniformly randomly either 2 or 3, all scaled by a factor of 1/24.

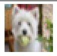
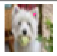
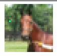
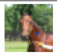
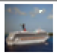
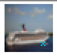
F.1.1 Comparison with other robust PCA methods on semi-synthetic data

In addition to comparing our results with simple pruning techniques, as we did in Figure 3 in the main text, we also compared our algorithm with implementations of other robust PCA techniques from the literature with accessible implementations. In particular, we compared our technique with RANSAC-based techniques, `LRVCov`, two SDPs ([CLMW11, XCS10]) for variants of robust PCA, and an algorithm proposed by [CLMW11] to speed up their SDP based on alternating descent. For the SDPs, since black box methods were too slow to run on the full data set (as [CLMW11] mentions, black-box solvers for the SDPs are impractical above perhaps 100 data points), we subsample the data, and run the SDP on the subsampled data. For each of these methods, we ran the algorithm on the true data points plus noise, where the noise was generated as described above. We then take the estimate of the covariance it outputs, and project the data points onto the top two singular values of this matrix, and plot the results in Figure F-1.

Similar results occurred for most noise patterns we tried. We found that only our algorithm and `LRVCov` were able to reasonably reconstruct Europe, in the presence of this noise. It is hard to judge qualitatively which of the two maps generated is preferable, but it seems that ours stretches the picture somewhat less than `LRVCov`.

F.2 Full table for watermarking experiments

Table F.1: Full table of accuracy and number of poisoned images left for different attack parameters. For each attack to target label pair, we provide a few experimental runs with different watermarks.

Sample	Target	Epsilon	Nat 1	Pois 1	# Pois Left	Nat 2	Pois 2	Std Pois
	bird	5%	92.27%	74.20%	57	92.64%	2.00%	1.20%
		10%	92.32%	89.80%	7	92.68%	1.50%	
	bird	5%	92.49%	98.50%	0	92.76%	2.00%	1.90%
		10%	92.55%	99.10%	0	92.89%	0.60%	
	bird	5%	92.66%	89.50%	14	92.59%	1.40%	1.10%
		10%	92.63%	95.50%	2	92.77%	0.90%	
	cat	5%	92.45%	83.30%	24	92.24%	0.20%	0.10%
		10%	92.39%	92.00%	0	92.44%	0.00%	
	cat	5%	92.60%	95.10%	1	92.51%	0.10%	0.10%
		10%	92.83%	97.70%	1	92.42%	0.00%	
	cat	5%	92.80%	96.50%	0	92.77%	0.10%	0.00%
		10%	92.74%	99.70%	0	92.71%	0.00%	
	dog	5%	92.91%	98.70%	0	92.59%	0.00%	0.00%
		10%	92.51%	99.30%	0	92.66%	0.10%	
	dog	5%	92.17%	89.80%	7	93.01%	0.00%	0.00%
		10%	92.55%	94.30%	1	92.64%	0.00%	
	horse	5%	92.38%	96.60%	0	92.87%	0.80%	0.80%
		10%	92.72%	99.40%	0	93.02%	0.40%	
	horse	5%	92.60%	99.80%	0	92.57%	1.00%	0.80%
		10%	92.26%	99.80%	0	92.63%	1.20%	
	cat	5%	92.68%	97.60%	1	92.72%	8.20%	7.20%
		10%	92.59%	99.00%	4	92.80%	7.10%	
	cat	5%	92.86%	98.60%	0	92.79%	8.30%	8.00%
		10%	92.29%	99.10%	0	92.57%	8.20%	
	deer	5%	92.68%	99.30%	0	92.68%	1.10%	1.00%
		10%	92.68%	99.90%	0	92.74%	1.60%	
	deer	5%	93.25%	97.00%	1	92.75%	2.60%	1.10%
		10%	92.31%	97.60%	1	93.03%	1.60%	
	frog	5%	92.87%	88.80%	10	92.61%	0.10%	0.30%
		10%	92.82%	93.70%	3	92.74%	0.10%	
	frog	5%	92.79%	99.60%	0	92.71%	0.20%	0.20%
		10%	92.49%	99.90%	0	92.58%	0.00%	
	bird	5%	92.52%	97.90%	0	92.69%	0.00%	0.00%
		10%	92.68%	99.30%	0	92.45%	0.50%	
	bird	5%	92.51%	87.80%	1	92.66%	0.20%	0.00%
		10%	92.74%	94.40%	0	92.91%	0.10%	

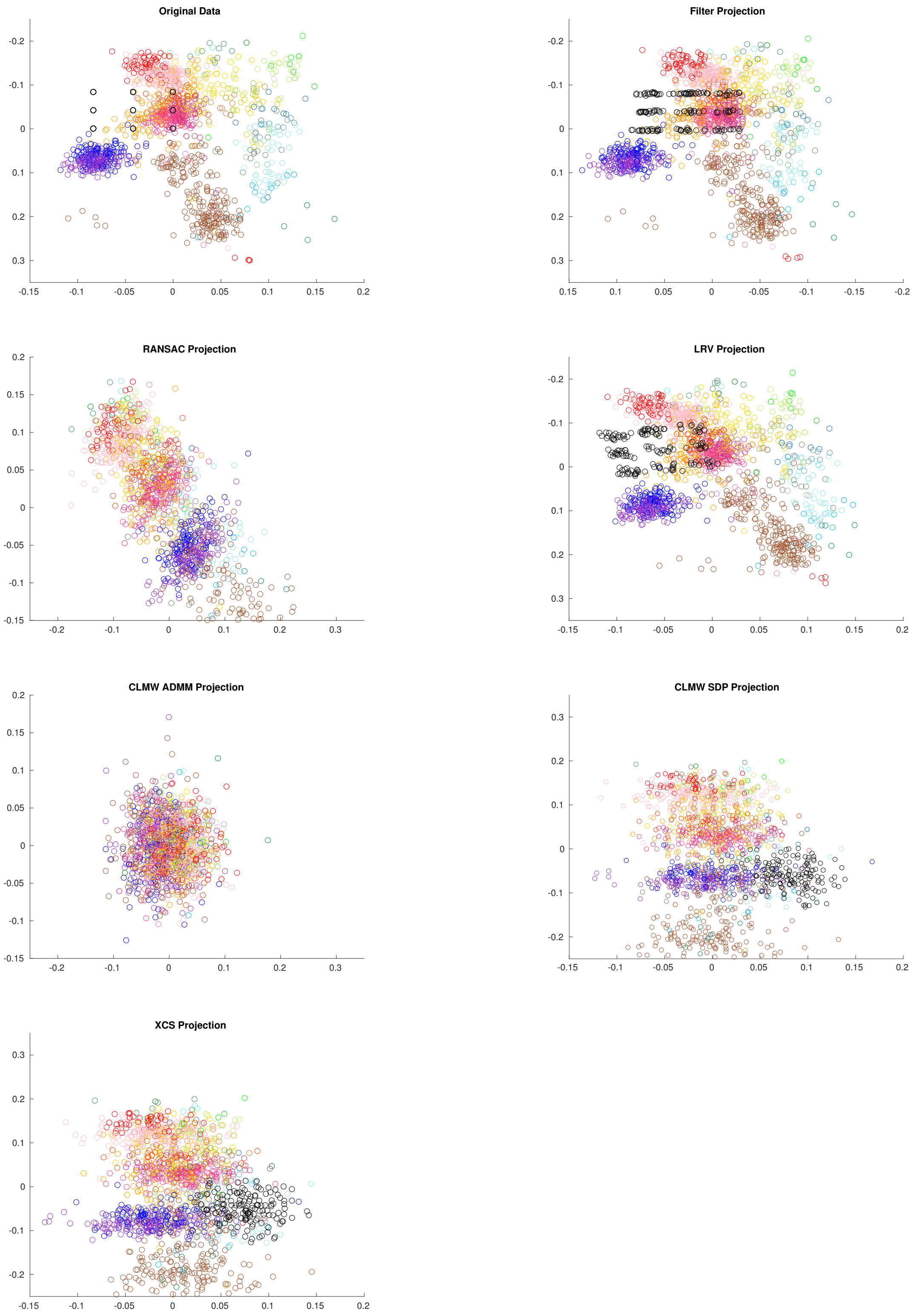


Figure F-1: Comparison with other robust methods on the Europe semi-synthetic data. From left to right, top to bottom: the original projection without noise, what our algorithm recovers, RANSAC, LRVCov, the ADMM method proposed by [CLMW11], the SDP proposed by [XCS10] with subsampling, and the SDP proposed by [CLMW11] with subsampling.

Appendix G

Additional Experimental Results for Sever

In this section, we provide additional plots of our experimental results, comparing with all baselines considered.

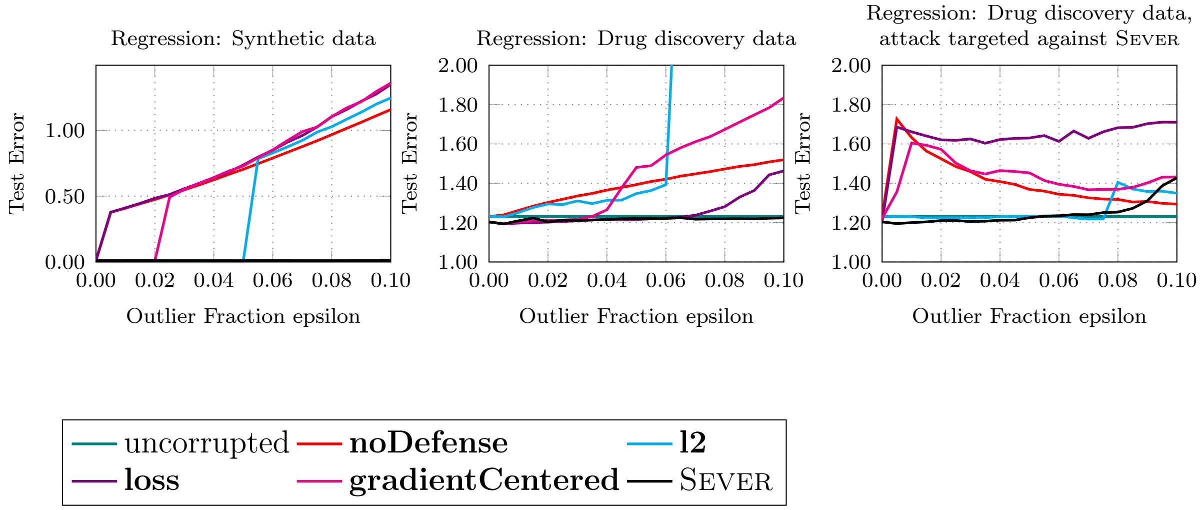


Figure G-1: ε vs test error for baselines and Sever on synthetic data and the drug discovery dataset. The left and middle figures show that Sever continues to maintain statistical accuracy against our attacks which are able to defeat previous baselines. The right figure shows an attack with parameters chosen to increase the test error Sever on the drug discovery dataset as much as possible. Despite this, Sever still has relatively small test error.

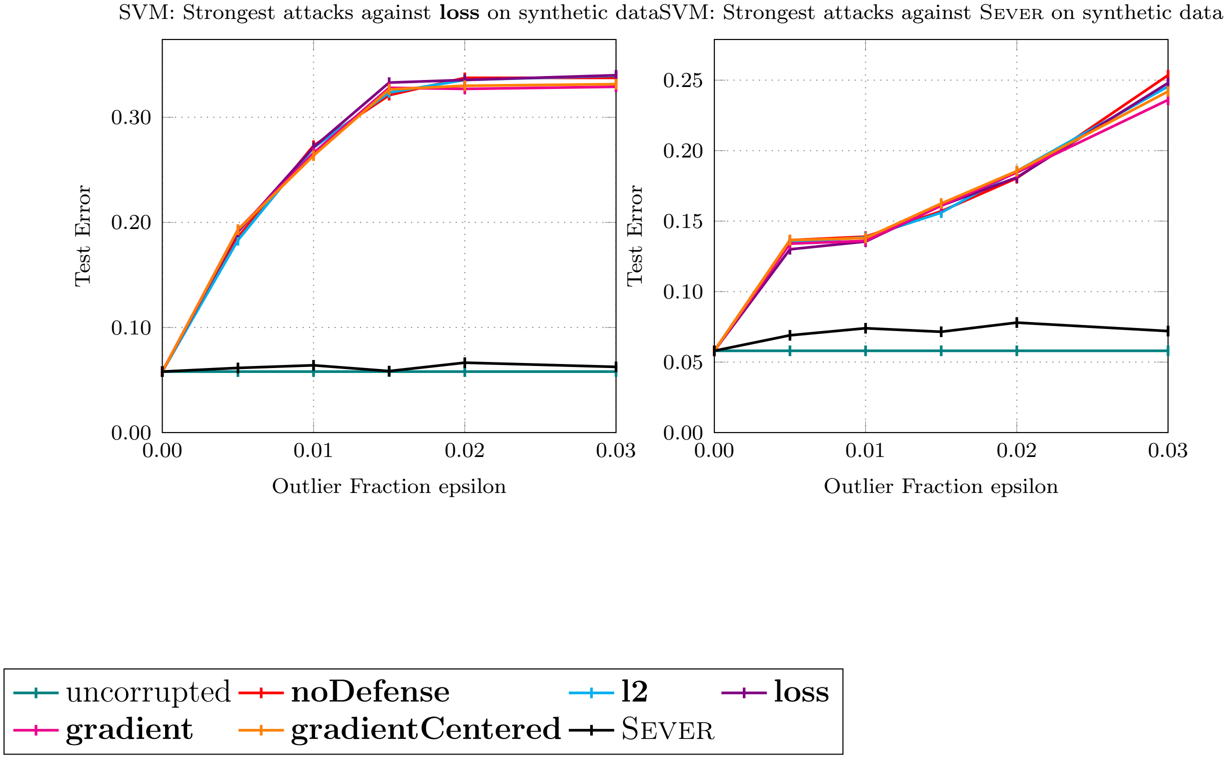


Figure G-2: ε vs test error for baselines and Sever on synthetic data. The left figure demonstrates that Sever is accurate when outliers manage to defeat previous baselines. The right figure shows the result of attacks which increased the test error the most against Sever. Even in this case, Sever performs much better than the baselines.

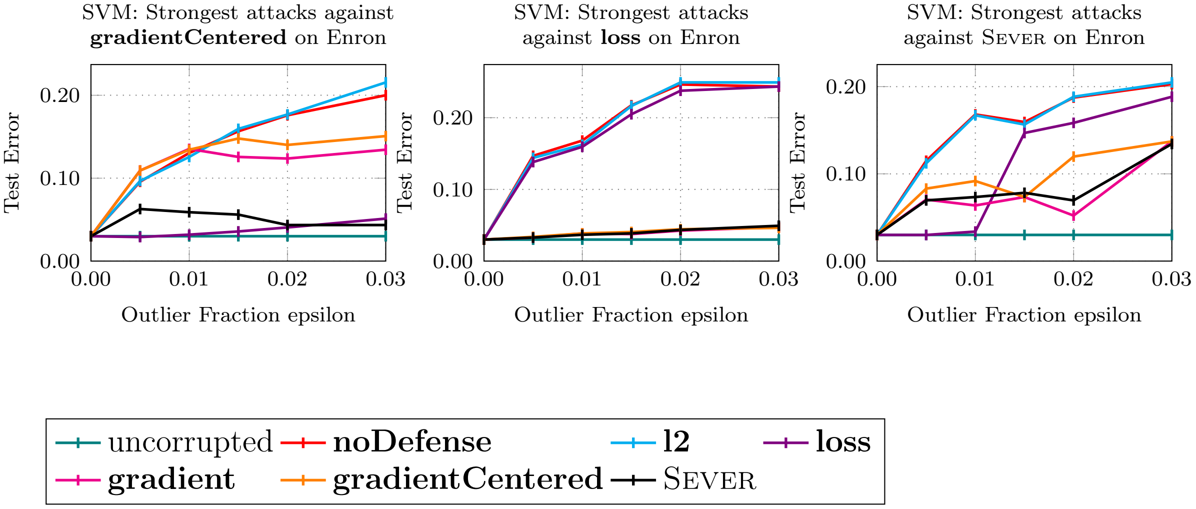


Figure G-3: ε versus test error for baselines and SEVER on the Enron spam corpus. The left and middle figures are the attacks which perform best against two baselines, while the right figure performs best against SEVER. Though other baselines may perform well in certain cases, only SEVER is consistently accurate. The exception is for certain attacks at $\varepsilon = 0.03$, which, as shown in Figure 7-6, require three rounds of outlier removal for any method to obtain reasonable test error – in these plots, our defenses perform only two rounds.

In this love scenario that we made
All the lights are now turned off
And when you flip the last page
The curtains will quietly fall—