Robustness meets algorithms

As Published: 10.1145/3453935

Publisher: Association for Computing Machinery (ACM)

Persistent URL: https://hdl.handle.net/1721.1/135599

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

DOI:10.1145/3453935

Robustness Meets Algorithms

By Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart

Abstract

In every corner of machine learning and statistics, there is a need for estimators that work not just in an idealized model, but even when their assumptions are violated. Unfortunately, in high dimensions, being provably robust and being efficiently computable are often at odds with each other.

We give the first efficient algorithm for estimating the parameters of a high-dimensional Gaussian that is able to tolerate a constant fraction of corruptions that is independent of the dimension. Prior to our work, all known estimators either needed time exponential in the dimension to compute or could tolerate only an inverse-polynomial fraction of corruptions. Not only does our algorithm bridge the gap between robustness and algorithms, but also it turns out to be highly practical in a variety of settings.

1. INTRODUCTION

Machine learning is filled with examples of estimators that work well in idealized settings but fail when their assumptions are violated. Consider the following illustrative example: We are given samples $X_1, X_2, \ldots, X_N$ from a one-dimensional Gaussian

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

and our goal is to estimate its mean μ and its variance σ^2. It is well-known that the empirical mean $\hat{\mu}$ and empirical variance $\hat{\sigma}^2$ are effective, which are defined as

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i \text{ and } \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \hat{\mu}\right)^2$$

In fact, these are examples of a more general paradigm within statistics called *maximum likelihood estimation*: When we know the distribution comes from some parametric family, we choose the parameters that are the most likely to have generated the observed data. In 1922, Ronald Fisher[12] formulated the maximum likelihood principle. It has many wonderful properties (under various technical conditions), such as converging to the true parameters as the number of samples goes to infinity, a property called *consistency*. Moreover, it has asymptotically the smallest possible variance among all unbiased estimators, a property called *asymptotic consistency*.

In 1960, John Tukey[24] challenged the conventional wisdom in parametric estimation by asking a simple question: Are there provably robust methods to estimate the parameters of a one-dimensional Gaussian? He showed that various estimators that were not asymptotically consistent (and had thus fallen out of favor) outperformed the maximum likelihood estimator when the data is not exactly Gaussian, but instead comes from a distribution that is close to being Gaussian. His paper launched the field of *robust statistics*[15, 13] that seeks to design estimators that behave well in a neighborhood around the true model. In one dimension, robust statistics prescribes that it is better to use the empirical median than the empirical mean. Similarly, it is better to use the empirical median absolute deviation (or any number of other estimators based on quantiles) than the empirical standard deviation. See Section 3.1.

Although there is an urgent need for provably robust estimators in virtually every application of machine learning, there is a major obstacle to directly applying ideas from robust statistics. The difficulty is that virtually all provably robust estimators are hard to compute in high dimensions. In this work, we are interested in the following family of questions:

QUESTION 1.1. *Let $\mathcal{D}$ be a family of distributions on $\mathbb{R}^d$. Suppose we are given samples generated from the following process: First, m samples are drawn from some unknown distribution P in $\mathcal{D}$. Then, an adversary is allowed to arbitrarily corrupt an ε-fraction of the samples. Can we efficiently find a distribution P' in $\mathcal{D}$ that is $f(\varepsilon, d)$-close, in total variation distance, to P?*

Our most important example is the direct generalization of John Tukey's challenge[24] to higher dimensions: Is there a provably robust *algorithm* for estimating the parameters of a high-dimensional Gaussian? Without algorithmic considerations, robust statistics already provides prescriptions such as the *Tukey median*[25] and the *minimum volume enclosing ellipsoid*[23] for estimating the high-dimensional mean and covariance, respectively. However, the best-known algorithms for computing these estimates run in time that is exponential in the dimension. In fact, we are not aware of any moderate-sized datasets with dimension larger than six where these estimates have been successfully computed!

In contrast, there are other techniques that one might try. For example, instead of computing the Tukey median, we could compute the coordinate-wise median. This can obviously be done in polynomial time but encounters a different sort of difficulty: It turns out that by adding corruptions along a direction that is not-axis aligned, an adversary can badly compromise the estimator. Quantitatively, even if an adversary is only allowed to corrupt only an ε-fraction of the samples, they can force the estimator to find a Gaussian that is as far as $\varepsilon\sqrt{d}$ in ℓ_2-distance, implying total variation distance is close to one.

The original version of this paper is entitled "Robust Estimators in High Dimensions without the Computational Intractability" and was published in *Proceedings of the 57th Annual IEEE Symp. on Foundations in Computer Science*. A version was also published in *SIAM J. on Computing*, 2019.

The main meta-question behind our work is: Is it possible to design estimators that are both provably robust in high dimensions (i.e., they do not lose dimension-dependent factors in their robustness guarantees) and also efficiently computable? We will give the first provably robust and computationally efficient methods for learning the parameters of a high-dimensional Gaussian, as well as various other related models. In concurrent and independent work, Lai et al.[20] gave alternative algorithms albeit with weaker guarantees. We discuss their work in Section 1.3.

The types of questions we study here also have roots in computational learning theory. They are related to the agnostic learning model of Kearns et al.[17] where the goal is to learn a labeling function whose agreement with some underlying target function is close to the best possible, among all functions in some given class. In contrast, we are interested in an unsupervised learning problem, but it is also agnostic in the sense that we want to find the approximately closest fit from among our family of distributions. Within machine learning, these types of problems are also called estimation under model misspecification. The usual prescription is to use the maximum likelihood estimator, which is unfortunately hard to compute in general. Even ignoring computational considerations, the maximum likelihood estimator is only guaranteed to converge to the distribution P' in $\mathcal{D}$ that is closest (in Kullback-Leibler divergence) to the distribution from which the observations are generated. This is problematic because such a distribution is not necessarily close to P at all.

More broadly, in recent years, there has been considerable progress on a variety of problems in this domain, such as algorithms with provable guarantees for learning mixture models, phylogenetic trees, hidden Markov models, topic models and independent component analysis. These algorithms are based on the method of moments and crucially rely on the assumption that the observations were actually generated by a model in the family. However, this simplifying assumption is not meant to be exactly true, and it is an important direction to explore what happens when it holds only in an approximate sense. Our work can be thought of as a first step toward relaxing the distributional assumptions in these applications, and subsequent work in algorithmic robust statistics has given new methodologies for robustly estimating higher moments under weaker assumptions about the uncorrupted distribution.[5, 10, 14, 19]

1.1. Our techniques

All of our algorithms are based on a common recipe. The first step is to answer the following easier question: Even if we were given a candidate hypothesis P', how could we tell if it is ε-close in total variation distance to P? The usual way to certify closeness is to exhibit a coupling between P and P' that marginally samples from both distributions, with the property that the samples are the same with probability $1 - \varepsilon$. However, we have no control over the process by which samples are generated from P, in order to produce such a coupling. And even then, the way that an adversary decides to corrupt samples can introduce complex statistical dependencies.

We circumvent this issue by working with an appropriate notion of parameter distance, which we use as a proxy for the total variation distance between two distributions in the class $\mathcal{D}$. See Section 2.2. Various notions of parameter distance underlie various efficient algorithms for distribution learning in the following sense. If θ and θ' are two sets of parameters that define distributions P_θ and $P_{\theta'}$ in a given class $\mathcal{D}$, a learning algorithm often relies on establishing the following type of relation between the total variation distance $d_{\mathrm{TV}}(P_\theta, P_{\theta'})$ and the parameter distance $d_p(\theta, \theta')$:

$$\mathrm{poly}(d_p(\theta,\theta'),1/d) \le d_{\mathrm{TV}}(P_\theta,P_{\theta'}) \le \mathrm{poly}(d_p(\theta,\theta'),d) \qquad (1)$$

Unfortunately, in our agnostic setting, we cannot afford for (1) to have any dependence on the dimension d at all. Any such dependence would appear in the error guarantee of our algorithm. Instead, the starting point of our algorithms is a notion of parameter distance that satisfies

$$\mathrm{poly}(d_p(\theta,\theta')) \le d_{\mathrm{TV}}(P_\theta,P_{\theta'}) \le \mathrm{poly}(d_p(\theta,\theta')) \qquad (2)$$

that allows us to reformulate our goal of designing robust estimators, with distribution-independent error guarantees, as the goal of robustly estimating θ according to d_p. In several settings, the choice of the parameter distance is rather straightforward. It is often the case that some variants of the ℓ_2-distance between the parameters work.

Given our notion of parameter distance satisfying (2), our main ingredient is an efficient method for robustly estimating the parameters. We provide two algorithmic approaches that are based on similar principles. Our first approach is fast and practical, requiring only approximate eigenvalue computations. Our second approach relies on convex programming, which has the advantage that it is possible to mix in different types of constraints (such as those generated by the sum-of-squares hierarchy) to tackle more complicated settings. Notably, either approach can be used to give all of our concrete learning applications with nearly identical error guarantees. In what follows, we specialize to the problem of robustly learning the mean μ of a Gaussian whose covariance is promised to be the identity, which we will use to illustrate how both approaches operate. We emphasize that learning the parameters in more general settings requires many additional ideas.

Our first algorithmic approach is an iterative greedy method that, in each iteration, filters out some of the corrupted samples. In particular, given a set of samples S' that contains a large set S of uncorrupted samples, an iteration of our algorithm either returns the sample mean of S' or finds a *filter* that allows us to efficiently compute a set $S'' \subset S'$ that is much closer to S. Note the sample mean $\hat{\mu} = \sum_{i=1}^{N} (1/N) X_i$ (even after we remove points that are obviously outliers) can be $\Omega(\varepsilon\sqrt{d})$-far from the true mean in ℓ_2-distance. The filter approach shows that either the sample mean is already a good estimate for μ or else there is an elementary spectral test that rejects some of the corrupted points and almost none of the uncorrupted ones. The crucial observation is that if a small number of corrupted points are responsible for a large change in the sample mean, it must be the case that many of the corrupted points are very far from the mean in some particular direction.

Our second algorithmic approach relies on convex programming. Here, instead of rejecting corrupted samples, we compute appropriate *weights* w_i for the samples X_i, so that the weighted empirical average $\hat{\mu}(w) \triangleq \sum_{i=1}^{N} \omega_i X_i$ is close to μ. We require the weights to be in the convex set C_τ, whose defining constraints are:

(a) $0 \le w_i \le \frac{1}{(1-\varepsilon)N}$ for all i and $\sum_{i=1}^{N} w_i = 1$, and

(b) $\left\| \sum_{i=1}^{N} w_i (X_i - \mu)(X_i - \mu)^T - I \right\|_2 \le \tau$.

We prove that *any* set of weights in $\mathcal{C}_\tau$ yields a good estimate $\hat{\mu}(w)$. The catch is that the set $\mathcal{C}_\tau$ is defined based on μ, *which is unknown*. Nevertheless, it turns out that we can use the same types of spectral arguments that underlie the filtering approach to design an approximate separation oracle for $\mathcal{C}_\tau$. Combined with standard results in convex optimization, this yields our second algorithm for robustly estimating μ.

The third and final ingredient is some new concentration bounds. In both of the approaches above, at best we hope that we can remove all of the corrupted points and be left with only the uncorrupted ones, and then use standard estimators (e.g., the empirical average) on them. However, an adversary could have removed an ε-fraction of the samples in a way that biases the empirical average of the remaining uncorrupted samples. What we need are concentration bounds that show for sufficiently large N, for samples $X_1, X_2, \ldots, X_N$ from a Gaussian with mean μ and identity covariance, that every $(1 - \varepsilon)N$ set of samples produces a good estimate for μ. In some cases, we can derive such concentration bounds by appealing to known concentration inequalities and taking a union bound. However, in other cases (e.g., concentration bounds for degree two polynomials of Gaussian random variables), the existing concentration bounds are not strong enough, and we need other arguments to prove what we need.

Finally, we briefly discuss how to adapt our techniques to robustly learn the covariance. Suppose the mean is zero and consider the following convex set $\mathcal{C}_\tau$, where Σ is the unknown covariance matrix:

(a) $0 \le w_i \le \frac{1}{(1-\varepsilon)N}$ for all i and $\sum_{i=1}^{N} w_i = 1$, and

(b) $\left\| \Sigma^{-1/2} \left(\sum_{i=1}^{N} w_i X_i X_i^T \right) \Sigma^{-1/2} - I \right\|_F \le \tau$.

Again, the constraints defining the convex set are based on the parameters of the distribution (this time, they use knowledge of Σ). We design an approximate separation oracle for this unknown convex set by analyzing the spectral properties of the fourth moment tensor of a Gaussian. It turns out that our algorithms for robustly learning the mean when the covariance is the identity and robustly learning the covariance when the mean is zero can be combined to solve the general problem.

1.2. Our results

We give the first efficient algorithms for agnostically learning several important distribution classes with dimension-independent error guarantees. Our main result is an algorithm for robustly learning a high-dimensional Gaussian with an almost optimal error guarantee. Throughout this paper, we write $N \ge \tilde{\Omega}(f(d, \varepsilon, \delta))$ when referring to our sample complexity, to signify that our algorithm works if $N \ge C\, f(d, \varepsilon, \delta)\mathrm{polylog}(f(d, \varepsilon, \delta))$ for a large enough universal constant C.

Theorem 1.2. *Let μ, Σ be arbitrary and unknown, and let $\varepsilon > 0$ be given. There is a polynomial time algorithm that given an ε-corrupted set of N samples from $\mathcal{N}(\mu, \Sigma)$ with $N \ge \tilde{\Omega}\left(\frac{d^3}{\varepsilon^2}\right)$ produces $\hat{\mu}$ and $\hat{\Sigma}$ so that with probability* 0.99, *we have*

$$d_{\mathrm{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \le O(\varepsilon \log^{3/2}(1/\varepsilon)).$$

In later work,[5] we improved the sample complexity to $N \ge \tilde{\Omega}\left(\frac{d^2}{\varepsilon^2}\right)$, which is optimal up to constant factors even when there are no corruptions. Moreover, it was observed in Diakonikolas et al.[6] that the error guarantee of these algorithms can be improved to $O(\varepsilon \log(1/\varepsilon))$, which is the best possible for statistical query algorithms.[9]

Beyond robustly learning a high-dimensional Gaussian, we give the first efficient robust learning algorithms with dimension-independent error guarantees for various other statistical tasks, such as robust estimation of a binary product distribution, robust density estimation for mixtures of any constant number of spherical Gaussians, and mixtures of two binary product distributions (under some natural balanced-ness condition). We emphasize that obtaining these results requires additional conceptual and technical ingredients. We defer a description of these results to the full version of our paper.

1.3. Related work

In concurrent and independent work, Lai et al.[20] also study high-dimensional robust estimation. Their results hold more generally for distributions with bounded moments, but our guarantees are stronger (and optimal up to polylogarithmic factors) for the fundamental problem of robustly learning a Gaussian.

After both our and their work, there has been a flurry of activity in the area, such as algorithms for robust list learning when the fraction of corruptions is greater than a half,[3] algorithms for sparse mean estimation whose sample complexity is sublinear in the dimension,[2] lower bounds against statistical query algorithms,[9] and extensions to other generative models with weaker moment conditions[14, 19] and various supervised learning problems.[22, 7] An overview of recent developments in the area can be found in Diakonikolas and Kane.[8] We also note that spectral techniques for robust learning, which are relatives of our algorithms, appeared in earlier work.[18, 1] These works employed a "hard" filtering step (for a supervised learning problem), which only removes outliers and consequently leads to errors that scale logarithmically with the dimension.

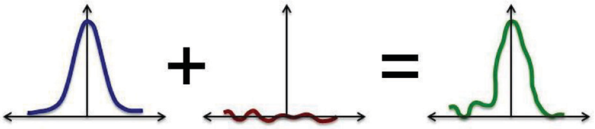
2. PRELIMINARIES

2.1. Problem setup

Formally, we will work in the following corruption model:

Definition 2.1. *For a given $\varepsilon > 0$ and an unknown distribution P, we say that S is an ε-corrupted set of samples from P of size N if $S = G \cup E \setminus S_r$, where G is a set of N independent samples from P, $S_r \subset G$, and E and S_r satisfy $|E| = |S_r| \le \varepsilon N$.*

In other words, a set of samples is ε-corrupted if an ε-fraction of the samples has been arbitrarily changed, which we can think of as a two-step process: first the adversary removes the samples in S_r and then adds in its own arbitrarily chosen data points E. Note that the ε-corruption model is a strong model of corruption, and it gives it more power than other classical notions of corruption, such as Huber's contamination model. We can visualize how the adversary can change the probability density function of P as follows:



Here, the blue curve is the original density function, and the green curve is the new density function, which is merely approximately close to P. The regions where the blue curve lies above the green curve are places where the adversary has deleted samples, and the regions where the green curve is above the blue curve are places where it has injected samples. In fact, the true process is even more complicated because if an adversary first inspects the samples and then decides what to corrupt, even the samples it has not corrupted are no longer necessarily independent.

It turns out that this model has very close connections to a natural measure of distance between distributions, namely, total variation distance:

Definition 2.2. *Given two distributions P, Q over $\mathbb{R}^d$ with probability density functions p, q, respectively, the* total variation *distance between P and Q is given by*

$$d_{\mathrm{TV}}(P,Q) = (1/2)\int_{\mathbb{R}^d} |\, p(x) - q(x)\,|\, dx.$$

The reason for this connection is as follows:

Fact 2.3. *Let P, Q be two distributions with $d_{\mathrm{TV}}(P, Q) = \varepsilon$. Let S be N i.i.d. samples from Q. Then, with probability at least $1 - \exp(-\Omega(\varepsilon n))$, S can be viewed as a set of $(1 + o(1))\varepsilon$-corrupted samples from P.*

This means that, up to subconstant factors in the fraction of corrupted points, learning from corrupted data is at least as hard as learning a distribution P from samples, if all we get are samples from some other distribution, which is ε-close to it in total variation distance. This fact immediately implies that if we are given ε-corrupted samples from P, the best we can generally hope for is to recover some $\hat{P}$ so that $d_{\mathrm{TV}}(P,\hat{P}) = \Theta(\varepsilon)$. As we shall see, it is often possible to match this lower bound (up to logarithmic factors).

2.2. Connections to parameter distance

In this paper, our focus is on *robust Gaussian estimation*, when $P = \mathcal{N}(\mu, \Sigma)$ is a Gaussian distribution. That is, given a set of N ε-corrupted samples from some unknown Gaussian distribution P, the goal is to output a Gaussian distribution $\hat{P}$ such that $d_{\mathrm{TV}}(P,\hat{P})$ is small. As it turns out, learning a Gaussian in total variation distance is closely tied to learning the parameters of the distribution, in the natural affine-invariant measure. This is captured by the following two lemmata. Throughout this paper, we will say that $f(x) \lesssim g(x)$ if $f(x) \le C_g(x)$ for all x and some universal constant C. We also let $\|A\|_F$ denote the Frobenius norm of a matrix A.

Lemma 2.4. *For any $\mu, \mu' \in \mathbb{R}^d$, we have*

$$\min(\|\mu - \mu'\|_2, 1) \lesssim d_{\mathrm{TV}}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I)) \lesssim \|\mu - \mu'\|_2 .$$

Lemma 2.5. *For any full-rank positive semidefinite matrices Σ, Σ', we have*

$$\min\left(\|\Sigma - \Sigma'\|_\Sigma, 1\right) \lesssim d_{\mathrm{TV}}(\mathcal{N}(0,\Sigma), \mathcal{N}(0,\Sigma')) \lesssim \|\Sigma - \Sigma'\|_\Sigma ,$$

where $\|A\|_\Sigma \triangleq \left\|\Sigma^{-1/2} A \Sigma^{-1/2}\right\|_F$.

The first lemma states that if the covariances are both the identity, then the total variation distance between the two Gaussians is essentially the ℓ_2-distance between the means of the Gaussians, except when the means are far apart. Note that total variation distance is always at most 1, and so when the means are very far apart, we only get a constant lower bound.

The second lemma is similar: It says that if both means are zero, then the total variation distance is captured by the Frobenius norm distance between the covariances, but "preconditioned" by one of the covariances. This is simply a high-dimensional analog of the fact that in one dimension, if we wish to get a meaningful approximation to the variance of a Gaussian, we need to learn it to multiplicative error.

3. ROBUST ESTIMATION

3.1. Univariate robust estimation

For the sake of exposition, we begin with robust univariate Gaussian estimation. A first observation is that the empirical mean is *not* robust: even changing a *single sample* can move our estimate by an arbitrarily large amount. To see this, let $\tilde{\mu}$ be the empirical mean of the dataset before corruptions, and let $\hat{\mu}$ be the empirical mean after increasing the value of the sample X_1 by some amount t. Although standard concentration arguments imply that $|\tilde{\mu} - \mu|$ is small, we have that $|\hat{\mu} - \tilde{\mu}| = t/N$, which we can make arbitrarily large with our choice of t. Fortunately, we describe a simple approach based on order statistics, which will allow us to estimate both the mean and the variance, even when a constant fraction of our dataset has been corrupted.

The most well-known robust estimator for the mean of a Gaussian is the *median*. More precisely, we let

$$\hat{\mu}_{\mathrm{med}} = \mathrm{median}(X_i).$$

Similarly, as an estimate for the standard deviation, we can consider a rescaling of the *median absolute deviation* (MAD), letting

$$\hat{\sigma}_{\text{mad}} = \frac{1}{\Phi^{-1}(3/4)} \cdot \text{median}(|X_i - \hat{\mu}|),$$

where Φ^{-1} is the inverse of the Gaussian cumulative distribution function. The rescaling is required to make the MAD a consistent estimator for the standard deviation. The median and MAD allow us to robustly estimate the underlying Gaussian:

THEOREM 3.1. *Given a set of* $N \geq \Omega\left(\frac{\log 1/\square}{\square^2}\right)$ *ε-corrupted samples from $\mathcal{N}(\mu, \sigma^2)$, with probability at least* $1 - \delta$, *we have*

$$d_{\text{TV}}\left(\mathcal{N}(\mu,\sigma^2), \mathcal{N}\left(\hat{\mu}_{\text{med}}, \hat{\sigma}^2_{\text{mad}}\right)\right) \leq C\varepsilon$$

for a universal constant C.

This estimator is the best of all possible worlds. It is provably robust. It can be computed efficiently. In fact, it also achieves the information-theoretically optimal sample complexity. The median and MAD are examples of robust estimators based on *order statistics*. There are other provably robust estimators based on *winsorizing*.[13]

3.2. Natural multivariate approaches that fail

There are many natural approaches for generalizing what we have learned in the one-dimensional case to the high-dimensional case. But as we will see, there is a tension between being provably robust and computationally efficient. First, consider a coordinate-by-coordinate approach where we robustly estimate the mean along each coordinate direction, and concatenate the d univariate estimates into an estimate for the d-dimensional mean vector. Although this achieves error $\Theta(\varepsilon)$ in each direction's subproblem, combining the estimates results in an ℓ_2 error of $\Omega(\varepsilon\sqrt{d})$. In high-dimensional settings, this gives vacuous bounds on the total variation distance except for vanishingly small values of ε.

Alternatively, one could attempt to extend the median-based estimator to multivariate settings. Although the same definition of the median does not apply in more than one dimension, there are many ways to generalize it. One such generalization is the *Tukey median*,[25] proposed specifically for the problem of robust estimation. The Tukey median of a dataset is the point (not necessarily in the dataset) that maximizes the minimum number of points on one side of any half-space through the point. Although this achieves the desired $O(\varepsilon)$ accuracy, it is unfortunately NP-hard to approximate on worst-case datasets.[16] Another multivariate notion of median is the *geometric median*, the point that minimizes the sum of ℓ_2-distances to points in the dataset. Although this is efficiently computable in polynomial time, it unfortunately also can be shown to incur $\Omega(\varepsilon\sqrt{d})$ error.[20]

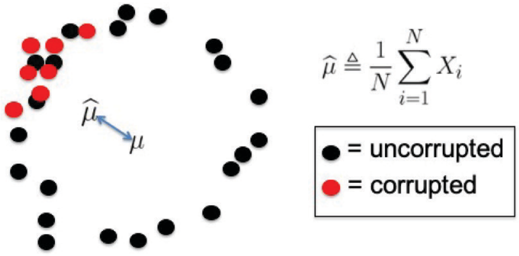
All the approaches mentioned so far have one of the following drawbacks:

1. The optimization problem is NP-hard, making it intractable in settings of even moderate dimensionality.
2. A large dimension-dependent factor appears in the error, resulting in very weak accuracy guarantees in high-dimensional settings.

At least one of these issues persists in all previously-known approaches, and either one would preclude realizable multivariate robust estimation. As few more examples, tournament-based hypothesis selection methods give accurate results, but are not computationally efficient. Alternatively, one could consider a pruning-based argument, which removes all points that are too far from the rest of the dataset. This is computationally efficient, but we again incur error of $\Omega(\varepsilon\sqrt{d})$. The primary contribution of our main result is a method that avoids both these issues simultaneously, providing an algorithm that is computationally efficient and does not lose a dimension-dependent factor in the accuracy.

3.3. Robust mean estimation

To offer some insight into why things go wrong in multivariate settings, we delve a bit deeper into the pruning-based approach. For the time being, we restrict our attention to Gaussians with identity covariance. It is well-known that, given a dataset generated according to a d-dimensional spherical Gaussian distribution, all the data points will be tightly concentrated at a distance $\Theta(\sqrt{d})$ from the mean. Thus, we can think about the distribution as being concentrated on a thin spherical shell, as depicted here:



A smart adversary can place all his corruptions within the shell too, in such a way that they move the empirical mean by as much as $\varepsilon\sqrt{d}$ in ℓ_2-distance. This demonstrates an intrinsic limitation of any algorithm, which only looks *locally* for corruptions—as a result, any effective algorithm must remove points based on *global* properties of the dataset.

This is captured in the following key geometric lemma:

LEMMA 3.2. *Let* $\varepsilon \in (0, 1/2)$. *Let S be an ε-corrupted set of points from $\mathcal{N}(\mu, I)$ of size at least $\Omega(d/\varepsilon^2)$. Let* $\hat{\mu}$, $\hat{\Sigma}$ *denote the empirical mean and covariance of S, that is,*

$$\hat{\mu} = \frac{1}{N}\sum_{X \in S} X, \qquad \hat{\Sigma} = \frac{1}{N}\sum_{X \in S}(X - \hat{\mu})(X - \hat{\mu})^{\top}.$$

Then, with probability at least 0.99, *we have:*

$$\|\mu - \hat{\mu}\|_2 \lesssim \varepsilon\sqrt{\log 1/\varepsilon} + \sqrt{\varepsilon\left\|\hat{\Sigma} - I\right\|_2}. \tag{3}$$

This lemma is a slight rephrasing of Lemma 4.15 in Diakonikolas et al.[4] At a high level, Lemma 3.2 states that if the true mean and the empirical (and potentially corrupted) mean are far apart, then the empirical variance must be noticeably different along some direction. Thus, the spectral norm of $\hat{\Sigma}$ can be used to certify that our estimate $\hat{\mu}$ is close to the true mean in the sense that

$$\|\bar{\Sigma} - I\|_2 = O(\varepsilon \log 1/\varepsilon) \Rightarrow \|\mu - \hat{\mu}\|_2 = O(\varepsilon\sqrt{\log 1/\varepsilon}).$$

On the other hand, when the empirical mean has been corrupted, Lemma 3.2 gives us a way to algorithmically make progress. It isolates a specific direction—namely, the top eigenvector of $\hat{\Sigma} - I$—in which the corrupted points must contribute a lot. Both of the algorithms we describe use information gleaned about the corruptions from the empirical moments in somewhat different ways.

Filtering approach. The filtering approach works by removing points from the dataset, using the above intuition. It proceeds as follows:

(a) Compute the top eigenvalue λ and eigenvector v of $\hat{\Sigma}$.
(b) If λ is sufficiently small, terminate and output $\hat{\mu}$.
(c) Otherwise, compute $\tau_i = \langle X_i - \hat{\mu}, v\rangle^2$, and for an adaptively chosen threshold T, remove all X_i so that $\tau_i > T$, and repeat.

If this is done carefully, then Lemma 3.2 guarantees that we always throw out many bad points compared to the number of good points we throw out. See Diakonikolas et al.[4] for a detailed description of how the threshold is chosen. To make this formal, for any two sets A, B, define $\Gamma(A,B) \triangleq |A \Delta B|/|A|$, which measures the relative size of the symmetric difference compared to the size of A. Then, we have the following guarantee for the filter:

LEMMA 3.3 (INFORMAL). *Let* $S = G \cup E \backslash S_r$ *be an* ε*-corrupted set of points from* $\mathcal{N}(\mu, I)$ *of size at least* $\tilde{\Omega}(d/\varepsilon^2)$. *Then, with probability at least* 0.99 *and after a simple preprocessing step, the filter satisfies the following property: Given any* $S' \subseteq S$ *satisfying* $\Gamma(G, S') \leq 2\varepsilon$, *the filter either*

(a) outputs $\hat{\mu}$ *so that* $\|\mu - \hat{\mu}\|_2 \leq O(\varepsilon\sqrt{\log 1/\varepsilon})$, *or*
(b) outputs T *so that* $\Gamma(G, T) \leq \Gamma(G, S') - \varepsilon/\alpha$, *where* $\alpha = d \log(d/\varepsilon) \log(d \log(d/\varepsilon))$.

Note that $\Gamma(G, S) \leq 2\varepsilon$ initially. Now applying Lemma 3.3 we can guarantee that the procedure terminates after at most $O(\alpha)$ iterations. And when we terminate, again by Lemma 3.3, we are guaranteed to output $\hat{\mu}$, which is close to the true mean. As described, each application of the filter would require computing the top eigenvector of a $d \times d$ matrix, which would be prohibitively slow. However, it turns out that a rough approximation to the top eigenvector suffices for the correctness of the filter algorithm, and thus each iteration can be performed in nearly-linear time via an approximate power method.

Convex programming approach. The second approach uses the intuition behind Lemma 3.2 somewhat differently. Instead of trying to directly remove all of the consequential outliers, we seek to iteratively decrease their influence. Let $S = G \cup E\backslash S_r$ be an ε-corrupted set of points. For each point $X_i \in S$, we associate a nonnegative weight w_i. Ideally, we would want these weights to be uniform over $S\backslash E$, and zero otherwise. *A priori* the only thing we know about $S\backslash E$ is that it has size at least $(1 - \varepsilon)N$. Consequently, a natural constraint to put on the weights w is that they must lie within the convex hull of the set of weights that are uniform on sets of size $(1 - \varepsilon)N$. This is the following set:

$$W_{n,\varepsilon} \triangleq \left\{ w \in \mathbb{R}^N : \sum_{i=1}^{N} w_i = 1, w_i \in \left[0, \frac{1}{(1-\varepsilon)N}\right] \right\}. \tag{4}$$

Given this, one can show that a slight extension of Lemma 3.2 implies that it suffices to find a set of weights $w \in W_{n,\varepsilon}$ so that the empirical distribution over S with these weights is spectrally close to the identity after centering at μ. To make this more formal, for any $w \in W_{N,\varepsilon}$, let

$$\mu(w) = \sum_{i=1}^{N} w_i X_i,\ \Sigma(w) = \sum_{i=1}^{N} w_i (X_i - \mu(w))(X_i - \mu(w))^\top$$

be the mean and covariance of the empirical distribution over S, with weights w. Define also

$$M(w) = \sum_{i=1}^{N} w_i (X_i - \mu)(X_i - \mu)^\top$$

to be the empirical covariance, except we center at the true mean of the unknown distribution. Note that $M(w)$ is a linear function of w, and so in particular, it is a convex function. Then, it suffices to solve the following convex problem:

$$\text{Find } w \in W_{N,\varepsilon} \text{ s.t. } \|M(w) - I\|_2 \leq C\varepsilon \log 1/\varepsilon, \tag{5}$$

where $C > 0$ is some universal constant. If $w \in W_{N,\varepsilon}$ satisfies (5), then one can show that $\mu(w)$ is close to μ with high probability, provided that $N = \Omega(d/\varepsilon^2)$.

There is an obvious difficulty in solving (5). Namely, the description of (5) requires knowledge of μ, the parameter that we wish to estimate! Luckily, we can still construct a separation oracle for (5), which will suffice for computing a solution. In particular, given $w \in W_{N,\varepsilon}$, we want an algorithm that

(a) if w satisfies (5), outputs YES and
(b) otherwise, outputs a hyperplane ℓ so that $\ell(w) > 0$ but $\ell(w') < 0$ for all w' satisfying (5).

First, note that if we knew μ, then constructing such an oracle would be straightforward. We simply compute the largest eigenvalue λ of $M(w) - I$ in magnitude, and its associated eigenvector v. If $|\lambda| < C\varepsilon \log 1/\varepsilon$, then output YES. If not, observe that the following is a separating hyperplane for w:

$$\sigma \cdot \left(\sum_{i=1}^{N} w_i \langle v, X_i - \mu \rangle^2 - 1 \right) > |\lambda|, \tag{6}$$

where σ is the sign of λ.

Now we need to remove the assumption that we know μ. The key insight is that Lemma 3.2 allows us to substitute the top eigenvalue of $\Sigma(w) - I$ in magnitude and its associated eigenvector for λ and v, respectively, and $\mu(w)$ for μ in (6). At a high level, this is because if $\mu(w)$ is close to μ, then $\Sigma(w)$ is very close to $M(w)$. On the other hand, if $\mu(w)$ is far from μ, then Lemma 3.2 guarantees that the shift caused by centering at $\mu(w)$ rather than μ is overshadowed by the large eigenvalues of $M(w) - I$.

3.4. Robust covariance estimation

The same geometric intuition that underlies our algorithms for robustly learning the mean also forms the basis for our algorithms for robustly learning the covariance. This time, we momentarily restrict our attention to Gaussians with zero mean. In the case of robust mean estimation, Lemma 3.2 states that a shift in the first moment caused by a small fraction of outliers causes a noticeable deviation in the second moment. It turns out that the same principles work for robustly learning the covariance, we just need to use higher moments. In particular, if we want to detect when the empirical second moment has been compromised by a small fraction of outliers, there must be some evidence in the fourth moment. However, making this rigorous is technically involved.

At a high level, the main difficulty is that in the case of robust mean estimation, we know the *structure* of the second moment, even if we do not know the mean. Namely, we assume that the covariance is the identity. However, the structure of the fourth moment depends heavily on the unknown covariance, and as a result, it is nontrivial to formulate the proper analog of Lemma 3.2 for this setting.

Fortunately, the relationship between the second moment and the fourth moment of a Gaussian follows a predictable formula, as a special case of *Isserlis' theorem*. For any vector $v \in \mathbb{R}^d$, let $v \otimes v \in \mathbb{R}^{d^2}$ denote the tensor product of v with itself. Similarly, for any matrix $M \in \mathbb{R}^{d \times d}$, let $M^{\otimes 2} \in \mathbb{R}^{d^2 \times d^2}$ be its tensor product with itself. Finally, let $M^\flat \in \mathbb{R}^{d^2}$ be the d^2-dimensional vector that comes from flattening M into a vector. Then, the key identity is the following: for any covariance matrix Σ, we have

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,\Sigma)}[(X \otimes X)(X \otimes X)^\top] = 2 \cdot \Sigma^{\otimes 2} + (\Sigma^\flat)(\Sigma^\flat)^\top. \tag{7}$$

Consider the case where the unknown covariance Σ is well-conditioned, so that it suffices to learn Σ to small error in Frobenius norm. Let $\{X_1, \ldots, X_N\}$ be an ε-corrupted dataset and set $Y_i = X_i \otimes X_i$ for all $i \in [N]$. If Y_i is uncorrupted, then $\mathbb{E}[Y_i] = \Sigma^\flat$, so recovering Σ in Frobenius norm exactly corresponds to learning the mean of the uncorrupted Y_i to small error in ℓ_2 norm. Moreover, by (7), the covariance of the uncorrupted Y_i is $2 \cdot \Sigma^{\otimes 2}$.

Thus learning the covariance reduces to a complicated variant of the mean estimation problem, where the covariance depends on the unknown mean, but in a structured way. The relationship between them is sufficiently nice so that if the empirical mean of the Y_i is corrupted by outliers, then this still manifests as a large eigenvalue of the empirical covariance of the Y_i. This allows us to formulate a more sophisticated analog of Lemma 3.2 for this setting (see Claim 4.29 in Diakonikolas et al.[4]). By then, leveraging this geometric structure, we can then devise generalizations of the filtering and the convex programming approaches to robustly learn the covariance.

3.5. Assembling the general algorithm

At this point, we have designed efficient algorithms to solve two important subcases of our general problem. Specifically, we can

(1) robustly estimate $\mathcal{N}(\mu, I)$, up to error $O\left(\varepsilon\sqrt{\log(1/\varepsilon)}\right)$ in total variation distance and
(2) robustly estimate $\mathcal{N}(0, \Sigma)$, up to error $O(\varepsilon \log(1/\varepsilon))$ in total variation distance.

In fact, we can combine these primitives into an algorithm that works in the general case when both μ and Σ are unknown. The first observation is that we can use the doubling trick (even in the presence of noise) to zero out the mean. In particular, given two independent samples X_1 and X_2 from a distribution that is ε-close to a Gaussian $\mathcal{N}(\mu, \Sigma)$, their difference $X_1 - X_2$ will be 2ε-close in distribution to $\mathcal{N}(0, 2\Sigma)$.

The second observation is that, given an estimate $\hat{\Sigma}$ of the covariance, we can approximately whiten our dataset. After applying the transformation $(\hat{\Sigma})^{-1/2}$ to our data, we get noisy samples from

$$\mathcal{N}(\hat{\Sigma}^{-1/2}\mu, \hat{\Sigma}^{-1/2}\Sigma\hat{\Sigma}^{-1/2}).$$

This *almost* fits into the setting where just the mean is unknown, because the resulting covariance matrix is close to (but not exactly equal to) the identity matrix. Fortunately, we can exploit the robustness of our algorithms to handle this error, because the data is generated from a distribution that is $O(\varepsilon \log(1/\varepsilon))$-close to a Gaussian with identity covariance. Putting the pieces together, we get an error guarantee of $O(\varepsilon \log^{3/2}(1/\varepsilon))$. The overall algorithm is described in Algorithm 1. We will use $\mathcal{X}$ and $\mathcal{Y}$ to denote a set of samples that are fed into various subroutines.

Algorithm 1 Algorithm for robustly learning a Gaussian

```
function RecoverRobustGaussian($\varepsilon, X_1, \ldots, X_{2N}$)
    Let $\hat{\Sigma} \leftarrow$ LearnCovariance($4\varepsilon, \mathcal{X}$) for
        $\mathcal{X} = (X_1 - X_2)/\sqrt{2}, \ldots, (X_{N-1} - X_N)/\sqrt{2}$,
    Let $\hat{\mu} \leftarrow$ LearnMean($O(\varepsilon \log(1/\varepsilon)), \mathcal{Y}$) for
        $\mathcal{Y} = \hat{\Sigma}^{-1/2} X_{N+1}, \ldots, \hat{\Sigma}^{-1/2} X_{2N}$
    return $\mathcal{N}(\hat{\Sigma}^{1/2}\hat{\mu}, \hat{\Sigma})$
```

4. EXPERIMENTS

Our algorithms (or rather, natural variants of them) not only have provable guarantees in terms of their efficiency and robustness but also turn out to be highly practical. In Diakonikolas et al.,[5] we studied their performance on both synthetic and real-world data, and we discuss the results in this section.

In Figure 1, we demonstrate our results on synthetic data, for estimating the mean and covariance of a Gaussian. We compare our Filtering method with the algorithms of Lai et al.,[20] the empirical plug-in estimator, the empirical estimator in combination with pruning, random sample consensus (RANSAC),[11] and the geometric median (for mean estimation). The first row of plots displays mean estimation for an isotropic Gaussian (with $\varepsilon = 0.1$), the second row displays covariance estimation for an isotropic Gaussian, and the third row displays covariance estimation for a Gaussian with a highly-skewed covariance matrix (both with $\varepsilon = 0.05$). The first column of plots compares all the methods, whereas the second column of plots omits the less accurate methods, to allow a more fine-grained comparison between our algorithm and competitive methods. The x-axis of each plot indicates the dimension of the problem, and the y-axis indicates the error incurred by the estimation method, where the baseline of 0 is the error of the plug-in estimator on the uncorrupted data. In Figure 1, for the mean estimation plots, this error is measured via ℓ_2-distance, whereas for covariance estimation, this is measured in terms of Mahalanobis distance.

Figure 1. Robust parameter estimation on synthetic data. Our method (filtering) is shown to outperform all alternatives for both mean estimation (first row) and covariance estimation (last two rows).

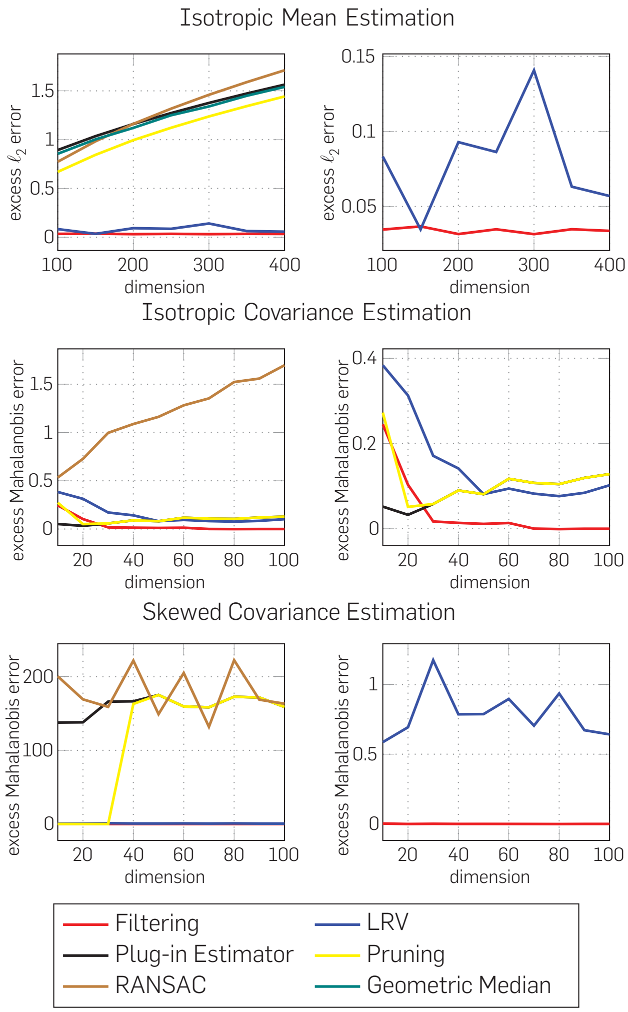
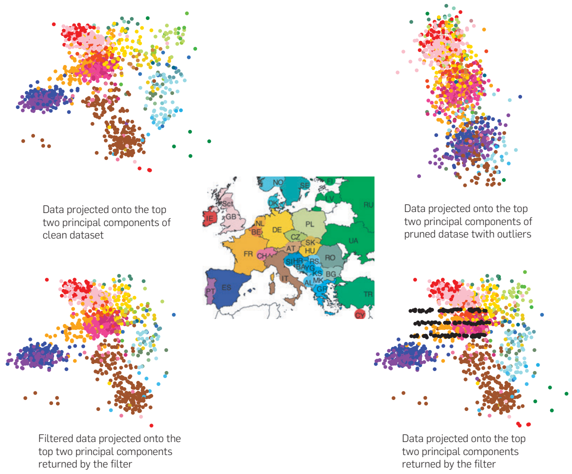


Figure 2. Robust exploratory data analysis on semisynthetic data. The top-left figure shows the projection of a high-dimensional genomic dataset onto its top two principal components, which resembles the map of Europe (center). In the presence of synthetic outliers, this structure is lost (top-right). Our methods for robust covariance estimation allow us to preserve this structure (bottom).


Data projected onto the top two principal components of clean dataset

Data projected onto the top two principal components of pruned datase twith outliers

Filtered data projected onto the top two principal components returned by the filter

Data projected onto the top two principal components returned by the filter

In all experiments, we found that our algorithm outperformed all other methods, often by substantial margins. As predicted by the theory, our error appears to remain constant as the dimension increases, but increases for all other methods (albeit minimally for LRV methods, which depend only logarithmically on the dimension). For mean estimation, our method performs better than LRV, which in turn performs much better than all alternatives. Similar trends are observed for covariance estimation, though the results are especially pronounced when estimating a skewed covariance, in which our methods outperform all others by orders of magnitude.

In our semisynthetic experiments, displayed in Figure 2, we revisit a classic study of Novembre et al.[21] In this study, the authors obtained a high-dimensional genomic dataset from the POPRES project. They annotated each data point with the individual's country of origin and projected the dataset onto the top two principal components of the dataset. As displayed in the top-left plot in Figure 2, they found that the resulting projection closely resembles the map of Europe, thus leading to the adage that "genes mirror geography." However, omitted from the description above is a crucial manual data curation process, in which immigrants were removed from the dataset, as they were considered to be genetic outliers. Our methods provide an automatic and principled way of removing outliers.

In our experiments, we worked with a projection of the original dataset onto the top 20 principal components. We injected synthetic noise points ($\varepsilon = 0.1$) into the dataset and repeated the experimental procedure described above. Even with a pruning step, we found the empirical estimator was not able to preserve the structure of Europe (top-right of Figure 1). However, our method (based on our robust Gaussian covariance estimation algorithm) was able to relatively faithfully recreate the original map of Europe (bottom-left and bottom-right of Figure 1). Despite our filter being designed for Gaussian data, the method worked on the genomic data (which is not necessarily Gaussian) with minimal alterations. C

References

1. Awasthi, P., Balcan, M.F., Long, P.M. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14 (New York, NY, USA, 2014), ACM, 449–458.
2. Balakrishnan, S., Du, S.S., Li, J., Singh, A. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Annual Conference on Learning Theory*, COLT '17 (2017), 169–212.
3. Charikar, M., Steinhardt, J., Valiant, G. Learning from untrusted data. In *Proceedings of the 49th Annual ACM Symposium on the Theory of Computing*, STOC '17 (New York, NY, USA, 2017), ACM, 47–60.
4. Diakonikolas, I., Kamath, G., Kane, D.M., Li, J., Moitra, A., Stewart, A. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '16 (Washington, DC, USA, 2016), IEEE Computer Society, 655–664.
5. Diakonikolas, I., Kamath, G., Kane, D.M., Li, J., Moitra, A., Stewart, A. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17 (2017), JMLR, Inc., 999–1008.
6. Diakonikolas, I., Kamath, G., Kane, D.M., Li, J., Moitra, A., Stewartz, A. Robustly learning a Gaussian: Getting optimal error, efficiently. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18 (Philadelphia, PA, USA, 2018), SIAM.
7. Diakonikolas, I., Kamath, G., Kane, D.M., Li, J., Steinhardt, J., Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In *Proceedings of the 36th International Conference on Machine Learning*, ICML '19 (2019), JMLR, Inc., 1596–1606.
8. Diakonikolas, I., Kane, D.M. Recent advances in algorithmic high-dimensional robust statistics. *CoRR*, abs/1911.05911, 2019.
9. Diakonikolas, I., Kane, D.M., Stewart, A. 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 (Washington, DC, USA, 2017), IEEE Computer Society, 73–84.
10. Diakonikolas, I., Kane, D.M., Stewart, A. List-decodable robust mean estimation and learning mixtures of spherical Gaussians. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18 (New York, NY, USA, 2018), ACM, 1047–1060.
11. Fischler, M.A., Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 6*, 24 (1981), 381–395.
12. Fisher, R.A. On the mathematical foundations of theoretical statistics. *Phil. Trans. R. Soc. Lond. Ser. A 594-604*, 222 (1922), 309–368.
13. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, Hoboken, New Jersey, 2011.
14. Hopkins, S.B., Li, J. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18 (New York, NY, USA, 2018), ACM, Hoboken, New Jersey, 1021–1034.
15. Huber, P.J., Ronchetti, E.M. *Robust Statistics*. Wiley, 2009.
16. Johnson, D.S., Preparata, F.P. The densest hemisphere problem. *Theor. Comp. Sci. 1*, 6 (1978), 93–107.
17. Kearns, M.J., Schapire, R.E., Sellie, L.M. Towards efficient agnostic learning. *Mach. Learn. 2–3*, 17 (1994), 115–141.
18. Klivans, A.R., Long, P.M., Servedio, R.A. Learning halfspaces with malicious noise. *J. Mach. Learn. Res.*, 10 (2009), 2715–2740.
19. Kothari, P., Steinhardt, J., Steurer, D. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18 (New York, NY, USA, 2018), ACM, 1035–1046.
20. Lai, K.A., Rao, A.B., Vempala, S. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '16 (Washington, DC, USA, 2016), IEEE Computer Society, 665–674.
21. Novembre, J., Johnson, T., Bryc, K., Kutalik, Z., Boyko, A.R., Auton, A., Indap, A., King, K.S., Bergmann, S., Nelson, M.R., Stephens, M., Bustamante, C.D. Genes mirror geography within Europe. *Nature* 7218, 456 (2008), 98–101.
22. Prasad, A., Suggala, A.S., Balakrishnan, S., Ravikumar, P. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485* (2018).
23. Rousseeuw, P. Multivariate estimation with high breakdown point. *Math. Statist. Appl.*, 8 (1985), 283–297.
24. Tukey, J.W. A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, Stanford University Press, Stanford, California, 1960, 448–485.
25. Tukey, J.W. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians* (1975), American Mathematical Society, 523–531.

Ilias Diakonikolas (ilias@cs.wisc.edu), University of Wisconsin, Madison, WI, USA.

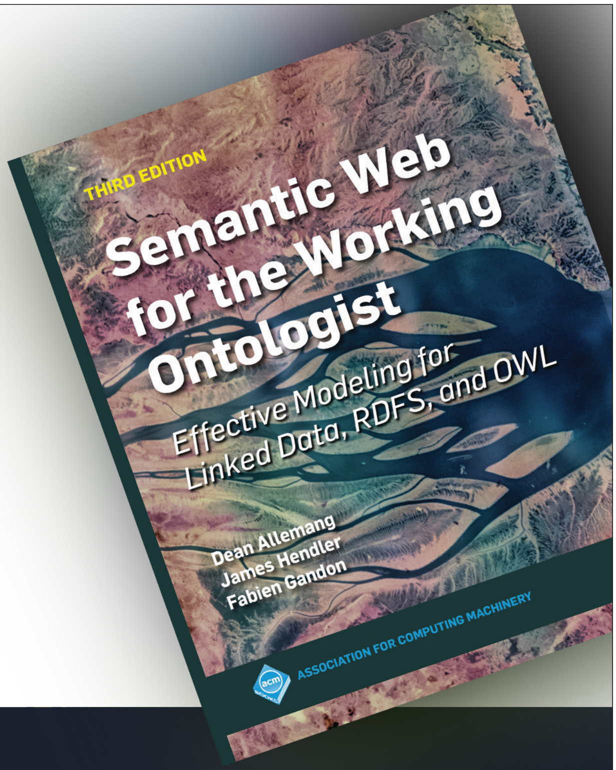
Gautam Kamath (g@csail.mit.edu), University of Waterloo, Canada.

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

Jerry Li (jerrl@microsoft.com), Microsoft Research AI, Redmond, WA, USA.

Ankur Moitra (moitra@mit.edu), Massachusetts Institute of Technology, Cambridge, MA, USA.

Alistair Stewart (stewart.al@gmail.com), Web3 Foundation, Zug, Switzerland.



THIRD EDITION
Semantic Web for the Working Ontologist
Effective Modeling for Linked Data, RDFS, and OWL
Dean Allemang
James Hendler
Fabien Gandon
ASSOCIATION FOR COMPUTING MACHINERY