Lecture 8: Additional topics in robust statistics

October 21, 2019

The study of learning given corrupted data is a very rich field of study, and we will not have time in this class to cover all the topics there. The goal of this lecture is simply to list a number of interesting directions that have recently received attention in this field, to give a number of good open questions in this area, and to also present the reader with sources where they can go into these topics in more detail, if interested.

1 Robust mean estimation via higher moments, sum-of-squares proofs, and lower bounds

Throughout this class so far, we have focused on finding deviations by looking at large directions of the second moment. This is very nice computationally, as we can find eigenvalues/eigenvectors of the second moment very efficiently via SVD / PCA and approximate versions thereof. However, from a statistical perspective, it is also natural to consider large directions of k-th moment tensors. Specifically, suppose we have a set of samples S_{good} with mean μ so that in all unit directions v, we have

$$\frac{1}{n}\sum_{i=1}^{n}\langle v, X_i - \mu\rangle^t \le C(t)\ , \tag{1}$$

for some (even) $t \ge 2$, and some function $C(t)$. Note that when $t = 2$ this exactly corresponds to S_{good} having bounded covariance in spectral norm. It is then not difficult to generalize the proof of the geometric lemma (Lemma 3.2 in Lecture 4) to demonstrate the following:

Lemma 1.1. *Let $S = S_{\text{good}} \cup S_{\text{bad}} \setminus S_r$ be ε-corrupted, so that S_{good} satisfies (1). Then, for all $w \in \mathscr{W}_{S,\varepsilon}$, we have*

$$\|\mu - \mu(w)\|_2 \le \frac{1}{1-\varepsilon}\left(C(t)^{1/t}\varepsilon^{1-1/t} + \lambda_t^{1/t}\varepsilon^{1-1/t}\right)\ ,$$

where

$$\lambda_t = \sup_{\|v\|_2=1}\frac{1}{n}\sum_{i\in S}\langle v, X_i - \mu(w)\rangle^t\ . \tag{2}$$

In other words, a large deviation in the empirical mean (asymptotically larger than $\varepsilon^{1-1/t}$) will manifest itself as a super-constant λ_t. This λ_t is the *injective tensor norm* of the empirical t-th moment tensor. It is not hard to see than when $t = 2$ we directly recover Lemma 3.2 from Lecture 4.

If we could recover the unit vector v which achieved the supremum in (2), then we could define scores

$$\tau_i = \langle v, X_i - \mu(w)\rangle^t\ , \tag{3}$$

and through similar arguments to what we did in Lecture 5, we could argue that the univariate filter makes progress until $\lambda_t = O(1)$, at which point Lemma 1.1 guarantees we have learned the true mean to error $O_t(\varepsilon^{1-1/t})$, where the $O_t(\cdot)$ hides factors that depend only on t. Unfortunately, the injective norm of a tensor is notoriously hard to compute in the worst case [1] under well-studied complexity assumptions. This

proof was simplified and adapted to this specific setting in [2] to demonstrate that even in this statistical setting, computing λ_t is likely hard. More directly, [2] gives some complexity-theoretic evidence that achieving $o(\varepsilon^{1/2})$ just from assuming (1) is hard, although from somewhat non-standard assumptions. A good open question is if the hardness of robust mean estimation can be derived from relatively standard assumptions such as the "Small Set Expansion" hypothesis.

Open Question 1.2. *Demonstrate that it is SSE-hard (or ETH-hard) to achieve $o(\varepsilon^{1/2})$ error while only assuming that the uncorrupted data satisfies* (1).

More ambitiously, one could ask the more general question of whether or not we can characterize which distributions allow for efficient robust mean estimation beyond $o(\varepsilon^{1/2})$. For instance, we know that we can do so for Gaussians with identity covariance, by leveraging this additional structure. However, this is a relatively strong distributional assumption.

We can somewhat interpolate between these extremes. Specifically, if we assume that the natural "Sum-of-squares" relaxation of (1) is satisfied, then there exist efficient estimators which achieve error $O_t(\varepsilon^{1-1/t})$ [3, 4] by lifting this problem into the sum-of-squares hierarchy. We will not define what these words mean (although they will come up repeatedly throughout this lecture!) since they are somewhat complicated to define. However, this assumption is fairly mild: [4] demonstrates that this relaxation holds for any distribution satisfying the so-called *Poincaré inequality*, which includes almost all natural distributions we like to consider, including for instance Gaussians. One notable exception for which we do not know if this works is the class of *log-concave* distributions. Unfortunately, establishing the Poincaré inequality for log-concave distributions is equivalent to the notorious KLS conjecture from convex geometry. This does not rule out the possibility of establishing the sum-of-squares relaxation of (1) directly for log-concave distributions, which would in turn establish a polynomial time algorithm for robust mean estimation for log-concave distributions beyond $\varepsilon^{1/2}$. This yields a nice open question:

Open Question 1.3. *Can we prove* (1) *in the sum-of-squares hierarchy, when the points X_i are drawn i.i.d. from a log-concave distribution? Alternatively, give any polynomial time algorithm for robust mean estimation for log-concave distributions which achieves rate $\varepsilon^{1-1/t}$, for $t > 2$.*

2 List-decodable learning

So far in this class we have universally assumed that $\varepsilon < 1/2$. In the classical sense of robust statistics, this is unavoidable. For instance, assume that I have an $1/2$-corrupted set of samples from $\mathcal{N}(\mu, I)$, and I wish to learn the mean μ. If the adversary then simply generates points distributed as $\mathcal{N}(\mu', I)$, then it is clearly impossible to choose between μ and μ' as the "correct" answer. More generally, suppose I have a $(1-\alpha)$-corrupted set of samples from $\mathcal{N}(\mu, I)$. Then, the adversary could choose $m = 1/\alpha - 1$ alternative means $\mu'_1, \dots, \mu'_{1/\alpha - 1}$, and make the corruptions be distributed as $\mathcal{N}(\mu'_i, I)$ with probability $1/m$. The resulting dataset is (roughly speaking) distributed as the uniform mixture over $\mathcal{N}(\mu, I), \mathcal{N}(\mu'_1, I), \dots, \mathcal{N}\,\mu'_m, I$, and so clearly we each one of the $\mu, \mu'_1, \dots, \mu'_m$ are equally valid-looking candidates to be the true mean.

However, one could hope that this is essentially the only way that the adversary can mess us up when $\varepsilon = 1 - \alpha$ is large (i.e. α is small). In particular, one could hope to output a succinct list of "candidate" means, so that at least one of these candidates is close to the true mean. This is the list-decodable learning problem, introduced by [5] in a slightly different setting, and [6] in this current form. The above instance says that the size of the list needs to be at least $\Omega(1/\alpha)$, and in fact this is achievable:

Theorem 2.1 ([6])**.** *Let D be a distribution with mean μ and covariance $\Sigma \preceq I$. Let S be an $(1-\alpha)$-corrupted set of samples from D of size $n \gtrsim \frac{d}{\alpha}$. Then, there is an algorithm which runs in time* $\mathrm{poly}(n, d, 1/\alpha)$ *which outputs a list of $m \lesssim 1/\alpha$ means $\mu_1, \dots, \mu_m$ so that $\|\mu - \mu_i\|_2 = \widetilde{O}(1/\sqrt{\alpha})$ for some $i \in [m]$, with probability* $1 - \exp(-\Omega(\alpha n))$.

As you'll show in the homework, the rate $O(1/\sqrt{\alpha})$ is statistically unavoidable for this problem, and so this gives the right answer up to logarithmic factors. Later work (partly using the Sum of Squares hierarchy) also showed how to achieve something similar when the distribution is Gaussian:

Theorem 2.2 ([7, 4]). *Let $t \geq 2$ be even. Let S be an $(1-\alpha)$-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size $n \gtrsim \mathrm{poly}(d, 1/\alpha)^t$. Then, there is an algorithm which runs in time $\mathrm{poly}(n, d, 1/\alpha)^t$ which outputs a list of $m \lesssim 1/\alpha$ means $\mu_1, \ldots, \mu_m$ so that $\|\mu - \mu_i\|_2 \lesssim f(t)\alpha^{-1/2t}$ for some $i \in [m]$, with probability at least $99/100$, and some function f that depends only on t.*

Another recent line of work has shown that regression tasks can also be performed in a list-decodable learning setting, see e.g. [8, 9].

3 Robust covariance estimation, robust distribution learning

So far in this class we have only really considered robust mean estimation. Another natural question is whether we can learn the covariance of the distribution, in the presence of errors. Let us consider the simplest form of this question, namely, learning the covariance of a mean-zero Gaussian, given outliers. It turns out this problem becomes significantly more involved, and indeed, will necessarily leverage a lot of the structure of Gaussian distributions.

The first step will be to establish what error metric we can hope to achieve. Recall from Lecture 2 this is dictated by how the total variation distance between two Gaussians depends on their covariance matrices. Ideally, given two covariance matrices Σ_1, Σ_2, we would like some metric $m(\cdot, \cdot)$ so that $\mathrm{d_{TV}}(\mathcal{N}(0, \Sigma_1), \mathcal{N}(0, \Sigma_2)) \approx m(\Sigma_1, \Sigma_2)$. We will not achieve this, but we will get close enough.

First, when $\Sigma_1 = I$, as you'll (partly) show in the homework, it turns out that the Frobenius norm distance between the two matrices essentially captures the total variation distance between the two matrices. Recall that the Frobenius norm of a symmetric matrix A, denoted $\|A\|_F$, is defined to be the square root of the sum of the squares of the entries of A, i.e. $\|A_F\| = \left(\sum A_{ij}^2\right)^{1/2}$, or equivalently, is the ℓ_2 norm of the vector of eigenvalues of A.

Fact 3.1. *We have that*

$$\min(1, \|I - \Sigma_2\|_F) \lesssim \mathrm{d_{TV}}(\mathcal{N}(0, I), \mathcal{N}(0, \Sigma_2)) \lesssim \|I - \Sigma_2\|_F \ .$$

To generalize this to arbitrary Σ_1, Σ_2, we now simply observe that total variation distance is rotationally invariant. That is, we have two distributions D_1, D_2, and if A is a (full-rank) linear transformation, and we let D_1' and D_2' be the distribution of AX_1 and AX_2, where $X_1 \sim D_1$ and $X_2 \sim D_2$ respectively, then $\mathrm{d_{TV}}(D_1, D_2) = \mathrm{d_{TV}}(D_1', D_2')$.

In our setting, assuming that Σ_1, Σ_2 are full-rank, we observe that if $X \sim \mathcal{N}(0, \Sigma_1)$, then $\Sigma_1^{-1/2} X \sim \mathcal{N}(0, I)$, where $\Sigma_1^{-1/2}$ is the matrix square-root, which is well-defined for full-rank PSD matrices. Moreover, if $X \sim \mathcal{N}(0, \Sigma_2)$, then $\Sigma_1^{-1/2} X \sim \mathcal{N}(0, \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2})$. Thus, we have that

$$\mathrm{d_{TV}}(\mathcal{N}(0, \Sigma_1), \mathcal{N}(0, \Sigma_2)) = \mathrm{d_{TV}}(\mathcal{N}(0, I), \mathcal{N}(0, \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2})) \ .$$

Combining this with the above fact, we have the following corollary:

Corollary 3.2. *Let Σ_1 be full-rank. We have that*

$$\min\left(1, \left\|I - \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2})\right\|_F\right) \lesssim \mathrm{d_{TV}}(\mathcal{N}(0, \Sigma_1), \mathcal{N}(0, \Sigma_2)) \lesssim \left\|I - \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}\right\|_F \ .$$

The quantity $\left\|I - \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}\right\|_F$ is known as the *Mahalanobis* distance between Σ_1, Σ_2 although strictly speaking it is not a distance as it is not symmetric. However, it simply acts as a "preconditioned" version of the Frobenius norm, when you precondition by the geometry of one of the covariance matrices. Indeed, if we define the norm $\|\cdot\|_\Sigma$ by $\|A\|_\Sigma = \left\|\Sigma^{-1/2} A \Sigma^{-1/2}\right\|_F$, then it is easy to see that

$$\left\|I - \Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}\right\|_F = \|\Sigma_1 - \Sigma_2\|_{\Sigma_1} \ .$$

This discussion motivates the following problem:

Problem 3.3. *Let $\Sigma \succ 0$, and let S be an ε-corrupted set of samples from $\mathcal{N}(0, \Sigma)$. Given $S, \varepsilon > 0$, output a $\widehat{\Sigma}$ so that with high probability, $\left\|\Sigma - \widehat{\Sigma}\right\|_\Sigma$ is minimized.*

Per the discussion above, one can show that information-theoretically, the best rate one can achieve for Problem 3.3 is $O(\varepsilon)$. The natural question is then whether we can match this rate efficiently. It is shown in [10] that this is in fact possible, up to logarithmic factors:

Theorem 3.4 ([10])**.** *Let ε, Σ, S be as in Problem 3.3, and suppose that $n \gtrsim O(d^2/\varepsilon^2)$, and assume that ε is sufficiently small. Then, there is an algorithm which runs in time* $\mathrm{poly}(n, d, 1/\varepsilon)$ *and which outputs $\widehat{\Sigma}$ so that with probability $\geq 99/100$, we have that $\left\|\Sigma - \widehat{\Sigma}\right\|_\Sigma \lesssim O(\varepsilon \log 1/\varepsilon)$.*

The approach will be the natural generalization of the approach we have taken for robust mean estimation. The key step will be again to establish some sort of spectral signatures lemma for this problem. At a high level, we will show that if our current estimate is off in Mahalanobis distance, then there will exist a large "eigenvalue" of the fourth moment. However, to get this right is a bit tricky, for two reasons. First off, as we've discussed already, eigenvalues of tensors are hard to compute. To get around this, we will instead work with a carefully chosen *flattening* of the fourth tensor from a $d \times d \times d \times d$-sized 4-tensor to a $d^2 \times d^2$-sized matrix. Conveniently, this turns out to be the natural object from a statistical perspective anyways.

The second issue is that, unlike for mean estimation, the structure of the fourth moment tensor can change as we change the (unknown) covariance. To handle this, we will need to use the fact that the fourth moment of a Gaussian has a nice form in terms of the second moment of the Gaussian, a fact known as *Isserlis' theorem*. This will allow us to compensate for this uncontrolled change in the fourth moment enough to still detect deviations to the fourth moment caused by the outliers. Unfortunately, this renders the algorithm heavily dependent on the algebraic structure of Gaussian moments (as Isserlis' theorem heavily depends on it). This appears to be unavoidable, however, as [4] demonstrates that we need to weaken our notion of recovery, for more general classes of distributions.

Learning arbitrary Gaussians robustly We've shown how to learn the mean of an isotropic Gaussian robustly, and also how to learn the covariance of a mean-zero Gaussian robustly. It turns out (if we're willing to lose some constant factors in ε) that these two primitives suffice to robustly learn an arbitrary Gaussian to total variation distance $O(\varepsilon \log 1/\varepsilon)$. You'll work through this in the homework.

3.1 Robust distribution learning

More generally, we can ask the following distribution learning question: suppose we have a class of distributions $\mathcal{D}$, and suppose we are given ε-corrupted samples from some $D \in \mathcal{D}$. Then, the goal is to output some $\widehat{D}$ so that $\mathrm{d_{TV}}(D, \widehat{D})$ is minimized. Information-theoretically, under mild assumptions on $\mathcal{D}$, error $O(\varepsilon)$ is possible, given enough samples. However, the computational question depends heavily on the structure of this distribution class. Many instances of this have been investigated in recent years. Beyond Gaussians, for instance, there has been work on robustly learning product distributions [10], mixtures of Gaussians [10], mixtures of product distributions [10], sparse Gaussian models [11], Ising models [12], batched univariate distributions [13], amongst many other settings. There is a natural open question in this line of work, of variable interest:

Open Question 3.5. *Let $\mathcal{D}$ be your favorite class of distributions. Given an ε-corrupted set of samples from an unknown $D \in \mathcal{D}$, give an efficient algorithm which outputs $\widehat{D}$ minimizing* $\mathrm{d_{TV}}(\widehat{D}, D)$.

A more ambitious, but likely impossible, question, is to give a broad characterization of what classes of distributions can be robustly and efficiently learned in the presence of outliers. This is related to the SSE-hardness-styles of questions presented above.

More generally, we can also consider other forms of corruption. Recall that learning from ε-corrupted data is almost equivalent to learning from ε-obliviously corrupted data, which simply means that I get

samples from some distribution D' which has small TV distance to the true D. However, there is no need to necessarily limit ourselves to perturbations in total variation distance. For instance, we could ask whether or not we can estimate D given samples from some distribution D' which has small earth-mover distance to D. A recent paper [14] study these sorts of questions in more depth, from an information-theoretic perspective. A great open question is whether or not the estimators in this paper can be made efficient.

4 Robust stochastic optimization

So far all of the problems considered have been unsupervised learning problems: we are given unlabeled data, and we are asked to recover some information about basic statistics of this data. However, a large part of modern machine learning is supervised learning, where we are given labeled data, and the goal is to recover the hidden relationship between the covariates and the labels. A natural question is whether we can also do supervised learning in the presence of adversarial corruptions to the training data.

The statistical setting that underlies these problems can be stated as follows. We assume there is some distribution D over covariate-response pairs (X, y), and some class of functions $\{f_\theta\}_{\theta \in C}$ parameterized by θ. We also assume there is some non-negative loss function ℓ which takes X, y, θ and outputs the loss of the model f_θ at the point (X, y). For instance, for regression $\ell(X, y, \theta) = \frac{1}{2}(\theta^\top X - y)^2$ is just the square loss. For every θ, we can define the expected loss of this model as

$$R(\theta) = \mathop{\mathbb{E}}_{(X,y)\sim D}[\ell(X, y, \theta)] \ .$$

In the typical (i.e. non-robust) setting, we simply assume we have a set of i.i.d. samples $(X_1, y_1), \ldots, (X_n, y_n)$ from D, and our goal is to recover $\theta^* = \arg\min R(\theta)$, that is, the model which minimizes the expected loss under D. Since this is typically impossible, we usually try to find some other θ with expected risk which is close to $R(\theta^*)$. A slightly weaker goal would be to find a point where the gradient of R is small, i.e. find θ so that $\|\nabla R(\theta)\|_2 \leq \sigma$. When R is convex, this corresponds to finding an approximate first order minimizer for the expected loss. Note that this setting is extremely general: it encompasses regression, logistic regression, SVM, and even learning deep nets.

We can generalize this problem somewhat: we can think of each (X, y) pair as actually generating a random function $f(\theta) = f_{X,y}(\theta) = \ell(X, y, \theta)$, and now we are actually receiving a set of random functions, drawn i.i.d. from some distribution P over functions. We can still define the expected function $R(\theta) = \mathbb{E}_{f\sim P}[f(\theta)]$, and now our goal is to, given an i.i.d. sample from P, to find some θ which approximately minimizes R. We can also consider the corresponding robust version of this problem:

Problem 4.1. *Let P be a probability distribution over functions $f : \mathbb{R}^d \to R$. Given an ε-corrupted set of samples from P, output some θ so that $\|\nabla R(\theta)\|_2$ is minimized.*

Notice that when f comes from the supervised learning setting, this corresponds to when an ε-fraction of (X_i, y_i) are arbitrarily corrupted. In the machine learning literature, this setting is often called the *data poisoning* setting.

How might one attack this problem? It turns out that here, we can directly use the technology we've developed for robust mean estimation, in an almost black-box manner. We will give two approaches for this problem, both using this technique, but in slightly different ways.

Robustifying SGD The first approach is to directly use robust mean estimation as a black-box, to robustify (stochastic) gradient descent. This was first introduced in [15, 16]. Recall that stochastic gradient descent is an iterative method which, given an iterate θ_t, applies the update

$$\theta_{t+1} = \theta_t - \eta g_t \ ,$$

where g_t is any random vector with $\mathbb{E}[g_t] = \nabla R(\theta_t)$. Given samples $f_1, \ldots, f_m \sim P$, a natural choice of g_t is given by the mini-batch gradient $g_t = \frac{1}{m}\sum_{i=1}^m \nabla f_i(\theta_t)$. It is well-known that, under reasonable assumptions

on P, SGD with the appropriate choice of step-size will converge to a point θ' so that $\|\nabla R(\theta')\|_2$ is small in expectation, and typically, the closer the estimate g_t is to the true $\nabla R(\theta_t)$, the faster the convergence.

If we only get ε-corrupted samples $S = S_{\text{good}} \cup S_{\text{bad}} \setminus S_r$ from P, we can no longer use the plug-in estimator for the gradient, as it will be typically far from $\nabla R(\theta_t)$. However, for $i \in S$, if we let $X_i = \nabla f_i(\theta_t)$, then observe that $\mathbb{E}[X_i] = \nabla R(\theta_t)$ for all $i \in S_{\text{good}}$. In other words, if we can robustly estimate the mean of the X_i, then we get a pretty decent estimate for $\nabla R(\theta_t)$, even given corrupted samples. There are many different assumptions made in convex analysis, but one relatively weak one is that the covariance of the gradients is bounded, i.e.

$$\mathop{\mathbb{E}}_{f\sim P}\left[(\nabla f(\theta) - R(\theta))\,(\nabla f(\theta) - R(\theta))^\top\right] \preceq I\ , \tag{4}$$

for all θ. But this is exactly the setting in which the filter algorithm gives good guarantees! Thus a natural algorithm is as follows: at every iteration, robustly estimate $\nabla R(\theta_t)$ from fresh corrupted samples to obtain some vector g_t, and update θ_t along this direction. While this will not produce an unbiased estimator for the gradient, it will produce one which is sufficiently close that one can still derive non-trivial convergence guarantees. Since the conditions under which this algorithm succeed and the actual guarantee is a bit tricky to state, we will refer the interested reader to [15, 16] for more details.

Robustifying a black-box learner The above approach has two main disadvantages. The first is that it only works with SGD. For many ML problems, there exist specialized solvers which use additional structure of the problem which are much faster than SGD, and so it would be advantageous if we could take any black-box learning algorithm for the problem, and somehow make it robust. The second is that it is quite slow. In particular, to get strong guarantees for the robust mean estimation step, we need to take a very large minibatch at every step of SGD. In the vanilla non-robust setting, typically we can take the batch size to be $m = O(1)$. However, to get strong guarantees for the robust mean estimation algorithm we perform at every iteration, we would need $m = \Omega(d/\varepsilon)$ at every iteration, and so we would run in time which is still at best $\Omega(d^2/\varepsilon)$ per iteration even with the nearly-linear time robust mean estimation algorithms. This represents a substantial slowdown, and renders this not so useful in practice.

To circumvent these difficulties, [16] proposes the following alternative approach.[1] Suppose we are given a black-box empirical risk minimizer, that is, an algorithm ERM which takes as input any set of functions S, and outputs θ so that $\theta = \arg\min \sum_{i\in S} f_i(\theta_i)$ (the arguments will also work with approximate ERM but we will not consider that for simplicity). Suppose furthermore that this ERM also allows minimizes weighted combinations of S, i.e. it takes as input (S, w), where $w \in \Gamma_S$, and outputs $\theta = \arg\min \sum_{i\in S} f_i(\theta_i)$. This assumption is somewhat non-standard but most ERM algorithms can easily be adapted to this setting as well.

The algorithm will proceed as follows: initially, maintain the uniform distribution over S, i.e. $w_0 = w(S)$. At iteration $t = 0, \ldots, T$, let $\theta_t = \text{ERM}(S, w_t)$. Then, for each $i \in S$, let $X_i = \nabla f_i(\theta_t)$, and apply one iteration of the filtering algorithm to these X_i to obtain new weights w_{t+1}. Formally, if the top eigenvalue of the covariance of the X_i is under some threshold, terminate and output θ_t. Otherwise, let τ_i be as if we were doing the filter on X_i, i.e.

$$\tau_i = \langle X_i - \widehat{\mu}, v\rangle^2\ ,$$

where $\widehat{\mu}$ is the average of the X_i and v is the top eigenvector of the covariance of the X_i, and let $w_{t+1} = \text{1DFilter}(\{X_1, \ldots, X_n\}, w_t)$, and we repeat.

We sketch at a high level why we might expect that this algorithm to succeed (handwaving away many details, see [16] for a full exposition). Recall that we proved that the filter satisfies the following invariant: in every iteration, either we terminate, and the empirical mean of the X_i with the current set of weights is close to the true mean of the X_i, or we remove more mass from the bad points than from the good points. If the filter terminates, i.e., then the filter guarantees that the empirical mean of the gradients is close to the true mean of the gradients. However, since we are filtering at the output of the ERM, we know the at the

[1] The algorithm presented there uses a randomized version of the filter, which does not maintain weights but rather removes points from the set S with some probability. However, the analyses of these two algorithms are essentially the same.

empirical mean of the gradients is 0. Then the filter guarantee implies that the true mean of the gradients is small, i.e., we are at an approximate local minima. On the other hand, if we do remove mass, then we remove more mass from outliers than inliers, and so we preserve the safety conditions of the algorithm. Since the filter always decreases the size of the support, we cannot run for too many iterations before we terminate.

While this approach does usually empirically perform better than the approach presented before, it is still far from optimal. In particular, for technical reasons it requires that S is quite large, and moreover, in the worst case the analysis can only prove that it requires $O(\varepsilon n)$ calls to the ERM oracle, as it is possible that it only removes one point from the support in each iteration. A great open question is whether these dependencies can be improved:

Open Question 4.2. *Give an algorithm which takes an ε-corrupted set of functions of size $n = \widetilde{O}(d/\varepsilon)$ from a distribution P satisfying (4), and outputs a θ so that $\|\nabla R(\theta)\|_2 = O(\sqrt{\varepsilon})$ in expectation, using only $\operatorname{poly}\log(n, d, 1/\varepsilon)$ calls to an ERM oracle.*

References

[1] Boaz Barak, Fernando GSL Brandao, Aram W Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 307–326. ACM, 2012.

[2] Samuel B Hopkins and Jerry Li. How hard is robust mean estimation? *arXiv preprint arXiv:1903.07870*, 2019.

[3] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034. ACM, 2018.

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

[5] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671–680. ACM, 2008.

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

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

[8] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. *arXiv preprint arXiv:1905.04660*, 2019.

[9] Sushrut Karmalkar, Pravesh Kothari, and Adam Klivans. List-decodable linear regression. *arXiv preprint arXiv:1905.05679*, 2019.

[10] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

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

[12] Yu Cheng, Ilias Diakonikolas, Daniel Kane, and Alistair Stewart. Robust learning of fixed-structure bayesian networks. In *Advances in Neural Information Processing Systems*, pages 10283–10295, 2018.

[13] Mingda Qiao and Gregory Valiant. Learning discrete distributions from untrusted batches. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[14] Banghua Zhu, Jiantao Jiao, and Jacob Steinhardt. Generalized resilience and robust statistics. *arXiv preprint arXiv:1909.08755*, 2019.

[15] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[16] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606, 2019.