Learning Halfspaces with Malicious Noise

Adam R. Klivans, Philip M. Long, and Rocco A. Servedio

klivans@cs.utexas.edu, plong@google.com, rocco@cs.columbia.edu

Abstract. We give new algorithms for learning halfspaces in the challenging *malicious noise* model, where an adversary may corrupt both the labels and the underlying distribution of examples. Our algorithms can tolerate malicious noise rates exponentially larger than previous work in terms of the dependence on the dimension n, and succeed for the fairly broad class of all isotropic log-concave distributions.

We give poly$(n, 1/\epsilon)$-time algorithms for solving the following problems to accuracy ϵ:

– Learning origin-centered halfspaces in $\mathbf{R}^n$ with respect to the uniform distribution on the unit ball with malicious noise rate $\eta = \Omega(\epsilon^2/\log(n/\epsilon))$. (The best previous result was $\Omega(\epsilon/(n\log(n/\epsilon))^{1/4})$.)

– Learning origin-centered halfspaces with respect to any isotropic log-concave distribution on $\mathbf{R}^n$ with malicious noise rate $\eta = \Omega(\epsilon^3/\log(n/\epsilon))$. This is the first efficient algorithm for learning under isotropic log-concave distributions in the presence of malicious noise.

We also give a poly$(n, 1/\epsilon)$-time algorithm for learning origin-centered halfspaces under any isotropic log-concave distribution on $\mathbf{R}^n$ in the presence of *adversarial label noise* at rate $\eta = \Omega(\epsilon^2/\log(1/\epsilon))$. In the adversarial label noise setting (or agnostic model), labels can be noisy, but not example points themselves. Previous results could handle $\eta = \Omega(\epsilon)$ but had running time exponential in an unspecified function of $1/\epsilon$.

Our analysis crucially exploits both concentration and anti-concentration properties of isotropic log-concave distributions. Our algorithms combine an iterative outlier removal procedure using Principal Component Analysis together with "smooth" boosting.

1 Introduction

A *halfspace* is a Boolean-valued function of the form $f = \text{sign}(\sum_{i=1}^{n} w_i x_i - \theta)$. Learning halfspaces in the presence of noisy data is a fundamental problem in machine learning. In addition to its practical relevance, the problem has connections to many well-studied topics such as kernel methods [26], cryptographic hardness of learning [13], hardness of approximation [5, 8], learning Boolean circuits [1], and additive/multiplicative update learning algorithms [15, 6].

Learning an unknown halfspace from correctly labeled (non-noisy) examples is one of the best-understood problems in learning theory, with work dating back to the famous Perceptron algorithm of the 1950s [21] and a range of efficient algorithms known for different settings [19, 14, 2, 17]. Much less is known, however, about the more difficult problem of learning halfspaces in the presence of noise.

Important progress was made by Blum *et al.* [1] who gave a polynomial-time algorithm for learning a halfspace under *classification noise*. In this model each label presented to the learner is flipped independently with some fixed probability; the noise does not affect the actual example points themselves, which are generated according to an arbitrary probability distribution over $\mathbf{R}^n$. In the current paper we consider a much more challenging noise model, *malicious noise*, which we describe below.

Malicious Noise. In this model, introduced by Valiant [27] (see also [12]), there is an unknown target function f and distribution $\mathcal{D}$ over examples. Each time the learner receives an example, independently with probability $1 - \eta$ it is drawn from $\mathcal{D}$ and labeled correctly according to f, but with probability η it is an arbitrary pair (x, y) which may be generated by an omniscient adversary. The parameter η is known as the "noise rate."

Malicious noise is a notoriously difficult model with few positive results. It was already shown in [12] that for essentially all concept classes, it is information-theoretically impossible to learn to accuracy $1-\epsilon$ if the noise rate η is greater than $\epsilon/(1+\epsilon)$. Indeed, known algorithms for learning halfspaces [25, 11] or even simpler target functions [18] with malicious noise typically make strong assumptions about the underlying distribution $\mathcal{D}$, and can learn to accuracy $1 - \epsilon$ only for noise rates η much smaller than ϵ.

In this paper we consider learning under the uniform distribution on the unit ball in $\mathbf{R}^n$, and more generally under any isotropic log-concave distribution. The latter is a fairly broad class of distributions that includes spherical Gaussians and uniform distributions over a wide range of convex sets. Our algorithms can learn from malicious noise rates that are quite high, as we now describe.

1.1 Main Results

Our first result is an algorithm for learning halfspaces in the malicious noise model with respect to the uniform distribution on the n-dimensional unit ball:

Theorem 1. *There is a* $\mathrm{poly}(n, 1/\epsilon)$*-time algorithm that learns origin-centered halfspaces to accuracy* $1 - \epsilon$ *with respect to the uniform distribution on the unit ball in* n *dimensions in the presence of malicious noise at rate* $\eta = \Omega(\epsilon^2/\log(n/\epsilon))$.

The previous best result is due to Kalai *et al.* [11] who gave a $\mathrm{poly}(n, 1/\epsilon)$-time algorithm for malicious noise at rate $\Omega(\epsilon/(n \log(n/\epsilon))^{1/4})$. Theorem 1 gives an exponential improvement in the dependence on n in the noise rate that can be achieved.

Via a more sophisticated algorithm, we can learn in the presence of malicious noise under any isotropic log-concave distribution:

Theorem 2. *There is a* $\mathrm{poly}(n, 1/\epsilon)$*-time algorithm that learns origin-centered halfspaces to accuracy* $1 - \epsilon$ *with respect to any isotropic log-concave distribution over* $\mathbf{R}^n$ *and can tolerate malicious noise at rate* $\eta = \Omega(\epsilon^3/\log(n/\epsilon))$.

We are not aware of any previous polynomial-time algorithms for learning under isotropic log-concave distributions in the presence of malicious noise.

Finally, we also consider a somewhat relaxed noise model known as *adversarial label noise*. In this model there is a fixed probability distribution P over $\mathbf{R}^n \times \{-1, 1\}$

(i.e. over labeled examples) for which a $1-\eta$ fraction of draws are labeled according to an unknown halfspace. The marginal distribution over $\mathbf{R}^n$ is assumed to be isotropic log-concave; so the idea is that an "adversary" chooses an η fraction of examples to mislabel, but unlike the malicious noise model she cannot change the (isotropic log-concave) distribution of the actual example points in $\mathbf{R}^n$. For this model we prove:

Theorem 3. *There is a* $\mathrm{poly}(n, 1/\epsilon)$*-time algorithm that learns origin-centered halfspaces to accuracy* $1-\epsilon$ *with respect to any isotropic log-concave distribution over* $\mathbf{R}^n$ *and can tolerate adversarial label noise at rate* $\eta = \Omega(\epsilon^2/\log(1/\epsilon))$.

The previous best algorithm for learning halfspaces in this framework, from [11] (where it is referred to as "agnostically learning halfspaces under log-concave distributions"), can tolerate $\eta = \Omega(\epsilon)$, but its running time is exponential in an unspecified function of $1/\epsilon$. We tolerate a somewhat lower noise rate (though one that is independent of n), but run in true $\mathrm{poly}(n, 1/\epsilon)$ time.

1.2 Techniques

Outlier Removal. Consider first the simplest problem of learning an origin-centered halfspace with respect to the uniform distribution on the n-dimensional ball. A natural idea is to use a simple "averaging" algorithm that takes the vector average of the positive examples it receives and uses this as the normal vector of its hypothesis halfspace. Servedio [24] analyzed this algorithm for the random classification noise model, and Kalai *et al.* [11] extended the analysis to the adversarial label noise model. [11] also showed that this simple algorithm learns to accuracy ϵ with malicious noise at rate less than $\epsilon/\sqrt{n}$. (Their $\Omega(\epsilon/(n\log(n/\epsilon))^{1/4})$ result was achieved via a different algorithm.)

Intuitively the "averaging" algorithm can only tolerate low malicious noise rates because the adversary can generate noisy examples which "pull" the average vector far from its true location. Our main insight is the adversary does this most effectively when the noisy examples are coordinated to pull in roughly the same direction. We use a form of outlier detection based on Principal Component Analysis to detect such coordination. This is done by computing the direction $\mathbf{w}$ of maximal variance of the data set; if the variance in direction $\mathbf{w}$ is suspiciously large, we remove from the sample all points $\mathbf{x}$ for which $(\mathbf{w}\cdot\mathbf{x})^2$ is large. Our analysis shows that this causes many noisy examples, and only a few non-noisy examples, to be removed.

We repeat this process until the variance in every direction is not too large. (This cannot take too many stages since many noisy examples are removed in each stage.) While some noisy examples may remain, we show that their disparate effects cannot hurt the algorithm much.

Thus, in a nutshell, our overall algorithm for the uniform distribution is to first do outlier removal[1] by an iterated PCA-type procedure, and then simply run the averaging algorithm on the remaining "cleaned-up" data set.

[1] We note briefly that the sophisticated outlier removal techniques of [1, 4] do not seem to be useful in our setting; those works deal with a strong notion of outliers, which is such that no point on the unit ball can be an outlier if a significant fraction of points are uniformly distributed on the unit ball.

Extending to Log-Concave Distributions via Smooth Boosting. We are able to show that the iterative outlier removal procedure described above is useful for isotropic log-concave distributions as well as the uniform distribution: if examples are removed in a given stage, then many of the removed examples are noisy and only a few are non-noisy (the analysis here uses concentration bounds for isotropic log-concave distributions). However, even if there were no noise in the data, the average of the positive examples under an isotropic log-concave distribution need not give a high-accuracy hypothesis. Thus the averaging algorithm alone will not suffice after outlier removal.

To get around this, we show that after outlier removal the average of the positive examples gives a (real-valued) *weak* hypothesis that has some nontrivial predictive accuracy. (Interestingly, the proof of this relies heavily on *anti*-concentration properties of isotropic log-concave distributions!) A natural approach is then to use a boosting algorithm to convert this weak learner into a strong learner. This is not entirely straightforward because boosting "skews" the distribution of examples; this has the undesirable effects of both increasing the effective malicious noise rate, and causing the distribution to no longer be isotropic log-concave. However, by using a "smooth" boosting algorithm [25] that skews the distribution as little as possible, we are able to control these undesirable effects and make the analysis go through. (The extra factor of ϵ in the bound of Theorem 2 compared with Theorem 1 comes from the fact that the boosting algorithm constructs "$1/\epsilon$-skewed" distributions.)

We note that our approach of using smooth boosting is reminiscent of [23, 25], but the current algorithm goes well beyond that earlier work. [23] did not consider a noisy scenario, and [25] only considered the averaging algorithm without any outlier removal as the weak learner (and thus could only handle quite low rates of malicious noise, at most $\epsilon/\sqrt{n}$ in our isotropic log-concave setting).

Finally, our results for learning under isotropic log-concave distributions with adversarial label noise are obtained using a similar approach. The algorithm here is in fact simpler than the malicious noise algorithm: since the adversarial label noise model does not allow the adversary to alter the distribution of the examples in $\mathbf{R}^n$, we can dispense with the outlier removal and simply use smooth boosting with the averaging algorithm as the weak learner. (This is why we get a slightly better quantitative bound in Theorem 3 than Theorem 2).

Organization. For completeness we review the precise definitions of isotropic log-concave distributions and the various learning models in Appendix A. We present the simpler and more easily understood uniform distribution analysis first, proving Theorem 1 in Section 2. The proof of Theorem 2, which builds on the ideas of Theorem 1, is in Section 3. Because of space constraints we prove Theorem 3 in Appendix C.

2 The uniform distribution and malicious noise

In this section we prove Theorem 1. As described above, our algorithm first does outlier removal using PCA and then applies the "averaging algorithm."

We may assume throughout that the noise rate η is smaller than some absolute constant, and that the dimension n is larger than some absolute constant.

2.1 The Algorithm: Removing Outliers and Averaging

Consider the following Algorithm A_{mu}:

1. Draw a sample S of $m = \mathrm{poly}(n/\epsilon)$ many examples from the malicious oracle.
2. Identify the direction $\mathbf{w} \in \mathbb{S}^{n-1}$ that maximizes

$$\sigma_{\mathbf{w}}^2 \stackrel{def}{=} \sum_{(\mathbf{x},y)\in S} (\mathbf{w}\cdot\mathbf{x})^2.$$

   If $\sigma_{\mathbf{w}}^2 < \frac{10m\log m}{n}$ then go to Step 4 otherwise go to Step 3.
3. Remove from S every example that has $(\mathbf{w}\cdot\mathbf{x})^2 \geq \frac{10\log m}{n}$. Go to Step 2.
4. For the examples S that remain let $\mathbf{v} = \frac{1}{|S|}\sum_{(\mathbf{x},y)\in S} y\mathbf{x}$ and output the linear classifier $h_{\mathbf{v}}$ defined by $h_{\mathbf{v}}(\mathbf{x}) = \mathrm{sgn}(\mathbf{v}\cdot x)$.

We first observe that Step 2 can be carried out in polynomial time:

Lemma 1. *There is a polynomial-time algorithm that, given a finite collection S of points in $\mathbf{R}^n$, outputs $\mathbf{w} \in \mathbb{S}^{n-1}$ that maximizes $\sum_{\mathbf{x}\in S}(\mathbf{w}\cdot\mathbf{x})^2$.*

Proof. If S is centered, i.e. $\sum_{\mathbf{x}\in S}\mathbf{x} = 0$, then the optimal $\mathbf{w}$ is the direction of maximum variance, and can be found using Principal Component Analysis (i.e. a polynomial-time eigenvector computation, see e.g. [10]). Otherwise, we can perform the PCA on $S \cup -S$, where $-S = \{-\mathbf{x} : \mathbf{x} \in S\}$. This works because $S \cup -S$ is centered, and, for each $\mathbf{w}$, $\sum_{\mathbf{x}\in S\cup -S}(\mathbf{w}\cdot\mathbf{x})^2 = 2\sum_{\mathbf{x}\in S}(\mathbf{w}\cdot\mathbf{x})^2$. □

This implies that the entire algorithm A_{mu} runs in $\mathrm{poly}(m)$ time.

Before embarking on the analysis we establish a terminological convention. Much of our analysis deals with high-probability statements over the draw of the m-element sample S; it is straightforward but quite cumbersome to explicitly keep track of all of the failure probabilities. Thus we write "with high probability" (or "w.h.p.") in various places below as a shorthand for "with probability at least $1 - 1/\mathrm{poly}(n/\epsilon)$." The interested reader can easily verify that an appropriate $\mathrm{poly}(n/\epsilon)$ choice of m makes all the failure probabilities small enough so that the entire algorithm succeeds with probability at least $1/2$ as required.

2.2 Properties of the clean examples

In this subsection we establish properties of the clean examples that were sampled in Step 1 of A_{mu}. The first says that no direction has much more variance than the expected variance of $1/n$:

Lemma 2. *W.h.p. over a random draw of ℓ clean examples S_{clean}, we have*

$$\max_{a\in\mathbb{S}^{n-1}} \left\{ \frac{1}{\ell} \sum_{(\mathbf{x},y)\in S_{\mathrm{clean}}} (\mathbf{a}\cdot\mathbf{x})^2 \right\} \leq \frac{1}{n} + \sqrt{\frac{O(n)\log m}{\ell}}.$$

Proof. The proof uses standard tools from VC theory and is in Appendix D. □

The next lemma says that in fact no direction has too many clean examples lying far out in that direction:

Lemma 3. *For any $\beta > 0$ and $\kappa > 1$, if S_{clean} is a random set of $\ell \geq \frac{O(1) \cdot n^2 \beta^2 e^{\beta^2 n/2}}{(1+\kappa)\ln(1+\kappa)}$ clean examples then w.h.p. we have*

$$\max_{a \in \mathbb{S}^{n-1}} \left\{ \frac{1}{\ell} \sum_{x \in S_{\text{clean}}} \mathbf{1}_{(a \cdot x)^2 > \beta^2} \right\} \leq (1+\kappa) e^{-\beta^2 n/2}.$$

Proof. In Appendix E. □

2.3 What is removed

In this section, we provide bounds on the number of clean and dirty examples removed in Step 3.

The first bound is a Corollary of Lemma 3.

Corollary 1. *W.h.p. over the random draw of the m-element sample S, the number of clean examples removed during any execution of Step 3 in A_{mu} is at most $6n \log m$.*

Proof. Since the noise rate η is sufficiently small, w.h.p. the number ℓ of clean examples is at least (say) $m/2$. We would like to apply Lemma 3 with $\kappa = 5\ell^4 n \log \ell$ and $\beta = \sqrt{\frac{10 \log m}{n}}$, and indeed we may do this because we have

$$\frac{O(1) \cdot n^2 \beta^2 e^{\beta^2 n/2}}{(1+\kappa)\ln(1+\kappa)} \leq \frac{O(1) \cdot n (\log m) m^5}{(1+\kappa)\ln(1+\kappa)} \leq O\left(\frac{m}{\log m}\right) \leq \frac{m}{2} \leq \ell$$

for n sufficiently large. Since clean points are only removed if they have $(\mathbf{a} \cdot \mathbf{x})^2 > \beta^2$, Lemma 3 gives us that the number of clean points removed is at most

$$m(1+\kappa)e^{-\beta^2 n/2} \leq 6m^5 n \log(\ell)/m^5 \leq 6n \log m.$$

□

The counterpart to Corollary 1 is the following lemma. It tells us that if examples are removed in Step 3, then there must be many *dirty* examples removed. It exploits the fact that Lemma 2 bounds the variance in *all* directions $\mathbf{a}$, so that it can be reused to reason about what happens in different executions of step 3.

Lemma 4. *W.h.p. over the random draw of S, whenever A_{mu} executes step 3, it removes at least $\frac{4m \log m}{n}$ noisy examples from S_{dirty}, the set of dirty examples in S.*

Proof. As stated earlier we may assume that $\eta \leq 1/4$. This implies that w.h.p. the fraction $\widehat{\eta}$ of noisy examples in the initial set S is at most $1/2$. Finally, Lemma 2 implies that $m = \tilde{\Omega}(n^2)$ suffices for it to be the case that w.h.p., for all $\mathbf{a} \in \mathbb{S}^{n-1}$, for the original multiset S_{clean} of clean examples drawn in step 1, we have

$$\sum_{(\mathbf{x},y) \in S_{\text{clean}}} (\mathbf{a} \cdot \mathbf{x})^2 \leq \frac{2m}{n}. \tag{1}$$

We shall say that a random sample S that satisfies all these requirements is "reasonable". We will show that for any reasonable dataset, the number of noisy examples removed during the execution of step 3 of A_{mu} is at least $\frac{4m \log m}{n}$.

If we remove examples using direction $\mathbf{w}$ then it means $\sum_{(\mathbf{x},y)\in S}(\mathbf{w}\cdot\mathbf{x})^2 \geq \frac{10m\log m}{n}$. Since S is reasonable, by (1) the contribution to the sum from the clean examples that survived to the current stage is at most $2m/n$ so we must have

$$\sum_{(\mathbf{x},y)\in S_{\mathrm{dirty}}}(\mathbf{w}\cdot\mathbf{x})^2 \geq 10m\log(m)/n - 2m/n > 9m\log(m)/n.$$

Let us decompose S_{dirty} into $N \cup F$ where N ("near") consists of those points x s.t. $(\mathbf{w}\cdot\mathbf{x})^2 \leq 10\log(m)/n$ and F ("far") is the remaining points for which $(\mathbf{w}\cdot\mathbf{x})^2 > 10\log(m)/n$. Since $|N| \leq |S_{\mathrm{dirty}}| \leq \widehat{\eta}m$, (any dirty examples removed in earlier rounds will only reduce the size of S_{dirty}) we have

$$\sum_{(\mathbf{x},y)\in N}(\mathbf{w}\cdot\mathbf{x})^2 \leq (\widehat{\eta}m)10\log(m)/n$$

and so

$$|F| \geq \sum_{(\mathbf{x},y)\in F}(\mathbf{w}\cdot\mathbf{x})^2 \geq 9m\log(m)/n - (\widehat{\eta}m)10\log(m)/n \geq 4m\log(m)/n$$

(the last line used the fact that $\widehat{\eta} < 1/2$). Since the points in F are removed in Step 3, the lemma is proved. □

2.4 Exploiting limited variance in any direction

In this section, we show that if all directional variances are small, then the algorithm's final hypothesis will have high accuracy.

We first recall a simple lemma which shows that a sample of "clean" examples results in a high-accuracy hypothesis for the averaging algorithm:

Lemma 5 ([24]). *Suppose $\mathbf{x}_1, ..., \mathbf{x}_m$ are chosen uniformly at random from $\mathbb{S}^{n-1}$, and a target weight vector $\mathbf{u} \in \mathbb{S}^{n-1}$ produces labels $y_1 = \mathrm{sign}(\mathbf{u}\cdot\mathbf{x}_1), ..., y_m = \mathrm{sign}(\mathbf{u}\cdot\mathbf{x}_m)$. Let $\mathbf{v} = \frac{1}{m}\sum_{t=1}^m y_t\mathbf{x}_t$. Then w.h.p. the component of $\mathbf{v}$ in the direction of $\mathbf{u}$ satisfies $\mathbf{u}\cdot\mathbf{v} = \Omega(\frac{1}{\sqrt{n}})$, while the rest of $\mathbf{v}$ satisfies $||\mathbf{v}-(\mathbf{u}\cdot\mathbf{v})\mathbf{u}|| = O(\sqrt{\log(n)/m})$.*

Now we can state Lemma 6.

Lemma 6. *Let $S = S_{\mathrm{clean}} \cup S_{\mathrm{dirty}}$ be the sample of m examples drawn from the noisy oracle $\mathrm{EX}_\eta(f,\mathcal{U})$. Let*

- *S'_{clean} be those clean examples that were never removed during step 3 of A_{mu},*
- *S'_{dirty} be those dirty examples that were never removed during step 3 of A_{mu},*
- *$\eta' = \frac{|S'_{\mathrm{dirty}}|}{|S'_{\mathrm{clean}}\cup S'_{\mathrm{dirty}}|}$, i.e. the fraction of dirty examples among the examples that survive step 3, and*
- *$\alpha = \frac{|S_{\mathrm{clean}} - S'_{\mathrm{clean}}|}{|S'_{\mathrm{clean}}\cup S'_{\mathrm{dirty}}|}$, the ratio of the number of clean points that were erroneously removed to the size of the final surviving data set.*

Let $S' \stackrel{def}{=} S'_{\mathrm{clean}} \cup S'_{\mathrm{dirty}}$*. Suppose that , for every direction* $\mathbf{w} \in \mathbb{S}^{n-1}$ *we have*

$$\sigma^2_{\mathbf{w}} \stackrel{def}{=} \sum_{(\mathbf{x},y)\in S'} (\mathbf{w}\cdot\mathbf{x})^2 \le \frac{10m\log m}{n}.$$

Then w.h.p. over the draw of S*, the halfspace with normal vector* $\mathbf{v} \stackrel{def}{=} \frac{1}{|S'|}\sum_{(\mathbf{x},y)\in S'} y\mathbf{x}$ *has error rate*

$$O\left(\sqrt{\eta'\log m} + \alpha\sqrt{n} + \sqrt{\frac{n\log n}{m}}\right).$$

Proof. The claimed bound is trivial unless $\eta' \le o(1)/\log m$ and $\alpha \le o(1)/\sqrt{n}$, so we shall freely use these bounds in what follows.

Let $\mathbf{u}$ be the unit length normal vector for the target halfspace. Let $\mathbf{v}_{\mathrm{clean}}$ be the average of *all* the clean examples, $\mathbf{v}'_{\mathrm{dirty}}$ be the average of the dirty (noisy) examples that were not deleted (i.e. the examples in S'_{dirty}), and $\mathbf{v}_{\mathrm{del}}$ be the average of the clean examples that were deleted. Then

$$\begin{aligned}
\mathbf{v} &= \frac{1}{|S'_{\mathrm{clean}}\cup S'_{\mathrm{dirty}}|}\sum_{(\mathbf{x},y)\in S'_{\mathrm{clean}}\cup S'_{\mathrm{dirty}}} y\mathbf{x} \\
&= \frac{1}{|S'_{\mathrm{clean}}\cup S'_{\mathrm{dirty}}|}\left(\left(\sum_{(\mathbf{x},y)\in S_{\mathrm{clean}}} y\mathbf{x}\right) + \left(\sum_{(\mathbf{x},y)\in S'_{\mathrm{dirty}}} y\mathbf{x}\right) - \left(\sum_{(\mathbf{x},y)\in S_{\mathrm{clean}}-S'_{\mathrm{clean}}} y\mathbf{x}\right)\right) \\
\mathbf{v} &= (1-\eta'+\alpha)\mathbf{v}_{\mathrm{clean}} + \eta'\mathbf{v}'_{\mathrm{dirty}} - \alpha\mathbf{v}_{\mathrm{del}}. \qquad (2)
\end{aligned}$$

Let us begin by exploiting the bound on the variance in every direction to bound the length of $\mathbf{v}'_{\mathrm{dirty}}$. For any $\mathbf{w}\in\mathbb{S}^{n-1}$ we know that

$$\sum_{(\mathbf{x},y)\in S'} (\mathbf{w}\cdot\mathbf{x})^2 \le \frac{10m\log m}{n}, \quad \text{and hence} \quad \sum_{(\mathbf{x},y)\in S'_{\mathrm{dirty}}} (\mathbf{w}\cdot\mathbf{x})^2 \le \frac{10m\log m}{n}$$

since $S'_{\mathrm{dirty}} \subseteq S'$. The Cauchy-Schwarz inequality now gives

$$\sum_{(\mathbf{x},y)\in S'_{\mathrm{dirty}}} |\mathbf{w}\cdot\mathbf{x}| \le \sqrt{\frac{10m|S'_{\mathrm{dirty}}|\log m}{n}}.$$

Taking $\mathbf{w}$ to be the unit vector in the direction of $\mathbf{v}'_{\mathrm{dirty}}$, we have $\|\mathbf{v}'_{\mathrm{dirty}}\| =$

$$\mathbf{w}\cdot\mathbf{v}'_{\mathrm{dirty}} = \mathbf{w}\cdot\frac{1}{|S'_{\mathrm{dirty}}|}\sum_{(\mathbf{x},y)\in S'_{\mathrm{dirty}}} y\mathbf{x} \le \frac{1}{|S'_{\mathrm{dirty}}|}\sum_{(\mathbf{x},y)\in S'_{\mathrm{dirty}}} |\mathbf{w}\cdot\mathbf{x}| \le \sqrt{\frac{10m\log m}{|S'_{\mathrm{dirty}}|n}}. \qquad (3)$$

Because the domain distribution is uniform, the error of $h_{\mathbf{v}}$ is proportional to the angle between $\mathbf{v}$ and $\mathbf{u}$, in particular,

$$\Pr[h_{\mathbf{v}} \ne f] = \frac{1}{\pi}\arctan\left(\frac{||\mathbf{v}-(\mathbf{v}\cdot\mathbf{u})\mathbf{u}||}{\mathbf{u}\cdot\mathbf{v}}\right) \le (1/\pi)\frac{||\mathbf{v}-(\mathbf{v}\cdot\mathbf{u})\mathbf{u}||}{\mathbf{u}\cdot\mathbf{v}}. \qquad (4)$$

We have that $||\mathbf{v} - (\mathbf{v} \cdot \mathbf{u})\mathbf{u}||$ equals

$$||(1 - \eta' + \alpha)(\mathbf{v}_{\text{clean}} - (\mathbf{v}_{\text{clean}} \cdot \mathbf{u})\mathbf{u}) + \eta'(\mathbf{v}'_{\text{dirty}} - (\mathbf{v}'_{\text{dirty}} \cdot \mathbf{u})\mathbf{u}) - \alpha(\mathbf{v}_{\text{del}} - (\mathbf{v}_{\text{del}} \cdot \mathbf{u})\mathbf{u})|| \\ \leq 2||\mathbf{v}_{\text{clean}} - (\mathbf{v}_{\text{clean}} \cdot \mathbf{u})\mathbf{u}|| + \eta'||\mathbf{v}'_{\text{dirty}}|| + \alpha||\mathbf{v}_{\text{del}}||$$

where we have used the triangle inequality and the fact that α, η are "small." Lemma 5 lets us bound the first term in the sum by $O(\sqrt{\log(n)/m})$, and the fact that $\mathbf{v}_{\text{del}}$ is an average of vectors of length 1 lets us bound the third by α. For the second term, Equation (3) gives us

$$\eta'\|\mathbf{v}'_{\text{dirty}}\| \leq \sqrt{\frac{10m(\eta')^2 \log m}{|S'_{\text{dirty}}|n}} = \sqrt{\frac{10m\eta' \log m}{|S'|n}} \leq \sqrt{\frac{20\eta' \log m}{n}},$$

where for the last equality we used $|S'| \geq m/2$ (which is an easy consequence of Corollary 1 and the fact that w.h.p. $|S_{\text{clean}}| \geq 3m/4$). We thus get

$$||\mathbf{v} - (\mathbf{v} \cdot \mathbf{u})\mathbf{u}|| \leq O\left(\sqrt{\log(n)/m}\right) + \sqrt{20\eta' \log(m)/n} + \alpha. \tag{5}$$

Now we consider the denominator of (4). We have

$$\mathbf{u} \cdot \mathbf{v} = (1 - \eta' + \alpha)(\mathbf{u} \cdot \mathbf{v}_{\text{clean}}) + \eta'\mathbf{u} \cdot \mathbf{v}'_{\text{dirty}} - \alpha\mathbf{u} \cdot \mathbf{v}_{\text{del}}.$$

Similar to the above analysis, we again use Lemma 5 (but now the lower bound $\mathbf{u} \cdot \mathbf{v} \geq \Omega(1/\sqrt{n})$, Equation (3), and the fact that $||\mathbf{v}_{\text{del}}|| \leq 1$. Since α and η' are "small," we get that there is an absolute constant c such that $\mathbf{u} \cdot \mathbf{v} \geq c/\sqrt{n} - \sqrt{20\eta' \log(m)/n} - \alpha$. Combining this with (5) and (4), we get

$$\Pr[h_{\mathbf{v}} \neq f] \leq \frac{O\left(\sqrt{\frac{\log n}{m}}\right) + \sqrt{\frac{20\eta' \log m}{n}} + \alpha}{\frac{c}{\sqrt{n}} - \sqrt{\frac{20\eta' \log m}{n}} - \alpha} = O\left(\sqrt{\frac{n \log n}{m}} + \sqrt{\eta' \log m} + \alpha\sqrt{n}\right).$$

□

2.5 Proof of Theorem 1

By Corollary 1, w.h.p. each outlier removal stage removes at most $6n \log m$ clean points.

Since each outlier removal stage removes at least $\frac{4m \log m}{n}$ noisy examples, there must be at most $O(n/(\log m))$ such stages. Consequently the total number of clean examples removed across all stages is $O(n^2)$. Since w.h.p. the initial number of clean examples is at least $m/2$, this means that the final data set (on which the averaging algorithm is run) contains at least $m/2 - O(n^2)$ clean examples, and hence at least $m/2 - O(n^2)$ examples in total. Consequently the value of α from Lemma 6 after the final outlier removal stage (the ratio of the total number of clean examples deleted, to the total number of surviving examples) is at most $\frac{2n^2}{m/2 - O(n^2)}$.

The standard Hoeffding bound implies that w.h.p. the actual fraction of noisy examples in the original sample S is at most $\eta + \sqrt{O(\log m)/m}$. It is easy to see that w.h.p.

the fraction of dirty examples does not increase (since each stage of outlier removal removes more dirty points than clean points, for a suitably large $\mathrm{poly}(n/\epsilon)$ value of m), and thus the fraction η' of dirty examples among the remaining examples after the final outlier removal stage is at most $\eta + \sqrt{O(\log m)/m}$. Applying Lemma 6, for a suitably large value $m = \mathrm{poly}(n/\epsilon)$, we obtain $\Pr[h_{\mathbf{v}} \neq f] \leq O\left(\sqrt{\eta \log m}\right)$. Rearranging this bound, we can learn to accuracy ϵ even for $\eta = \Omega(\epsilon^2/\log(n/\epsilon))$. This completes the proof of the theorem. □

3 Isotropic log-concave distributions and malicious noise

Our algorithm A_{mlc} that works for arbitrary log-concave distributions uses smooth boosting.

3.1 Smooth Boosting

A boosting algorithm uses a subroutine, called a *weak learner*, that is only guaranteed to output hypotheses with a non-negligible advantage over random guessing.[2] The boosting algorithm that we consider uses a *confidence-rated* weak learner [22], which predicts $\{-1, 1\}$ labels using continuous values in $[-1, 1]$. Formally, the *advantage* of a hypothesis h' with respect to a distribution $\mathcal{D}'$ is defined to be $\mathbf{E}_{x \sim \mathcal{D}'}[h'(x) f(x)]$, where f is the target function.

For the purposes of this paper, a boosting algorithm makes use of the weak learner, an example oracle (possibly corrupted with noise), a desired accuracy ϵ, and a bound γ on the advantage of the hypothesis output by the weak learner.

A boosting algorithm that is trying to learn an unknown target function f with respect to some distribution $\mathcal{D}$ repeatedly simulates a (possibly noisy) example oracle for f with respect to some other distribution $\mathcal{D}'$ calls a subroutine A_{weak} with respect to this oracle, receiving a *weak hypothesis*, which maps $\mathbf{R}^n$ to the continuous interval $[-1, 1]$.

After repeating this for some number of stages, the boosting algorithm combines the weak hypotheses generated durings its various calls to the weak learner into a final aggregate hypothesis which it outputs.

Let $\mathcal{D}, \mathcal{D}'$ be two distributions over $\mathbf{R}^n$. We say that $\mathcal{D}'$ is $(1/\epsilon)$-*smooth with respect to* $\mathcal{D}$ if $\mathcal{D}(\mathbf{x}) \leq (1/\epsilon)\mathcal{D}'(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{R}^n$.

The following lemma from [25] (similar results can be readily found elsewhere, see e.g. [7]) identifies the properties that we need from a boosting algorithm for our analysis.

Lemma 7 ([25]). *There is a boosting algorithm B and a polynomial p such that, for any $\epsilon, \gamma > 0$, the following properties hold. When learning a target function f using $\mathrm{EX}_\eta(f, \mathcal{D})$, we have: (a) If each call to A_{weak} takes time t, then B takes time $p(t, 1/\gamma, 1/\epsilon)$. (b) The weak learner is always called with an oracle $\mathrm{EX}_{\eta'}(f, \mathcal{D}')$ where $\mathcal{D}'$ is $(1/\epsilon)$-smooth with respect to $\mathcal{D}'$ and $\eta' \leq \eta/\epsilon$. (c) Suppose that for each distribution $\mathrm{EX}_{\eta'}(f, \mathcal{D}')$ passed to A_{weak} by B, the output of A_{weak} has advantage γ. Then the final output h of B satisfies $\Pr_{x \in \mathcal{D}}[h(x) \neq f(x)] \leq \epsilon$.*

[2] For simplicity of presentation we ignore the confidence parameter of the weak learner in our discussion; this can be handled in an entirely standard way.

3.2 The Algorithm

Our algorithm for learning under isotropic log-concave distributions with malicious noise, Algorithm A_{mlc}, applies the smooth booster from Lemma 7 with the following weak learner, which we call Algorithm A_{mlcw}. (The value c_0 is an absolute constant that will emerge from our analysis.)

1. Draw $m = \mathrm{poly}(n/\epsilon)$ examples from the oracle $\mathrm{EX}_{\eta'}(f, \mathcal{D}')$.
2. Remove all those examples $(\mathbf{x}, y)$ for which $||\mathbf{x}|| > \sqrt{3n \log m}$.
3. Repeatedly
   - find a direction (unit vector) $\mathbf{w}$ that maximizes $\sum_{(\mathbf{x},y)\in S}(\mathbf{w} \cdot \mathbf{x})^2$ (see Lemma 1)
   - if $\sum_{(\mathbf{x},y)\in S}(\mathbf{w}\cdot\mathbf{x})^2 \le c_0 m \log(n/\epsilon)$ then move on to Step 3, and otherwise
   - remove from S all examples $(\mathbf{x}, y)$ for which $(\mathbf{w} \cdot \mathbf{x})^2 > c_0 \log(n/\epsilon)$, and iterate again.
4. Let $\mathbf{v} = \frac{1}{|S|}\sum_{(\mathbf{x},y)\in S} y\mathbf{x}$, and return h defined by $h(\mathbf{x}) = \frac{\mathbf{v}\cdot\mathbf{x}}{3n \log m}$, if $|\mathbf{v} \cdot \mathbf{x}| \le 3n \log m$, and $h(\mathbf{x}) = \mathrm{sgn}(\mathbf{v} \cdot \mathbf{x})$ otherwise.

Our main task is to analyze the weak learner. Given the following Lemma, Theorem 2 will be an immediate consequence of Lemma 7 (proved in Appendix B).

Lemma 8. *Suppose Algorithm A_{mlcw} is run using $\mathrm{EX}_{\eta'}(f, \mathcal{D}')$ where f is an origin-centered halfspace, $\mathcal{D}'$ is $(1/\epsilon)$-smooth w.r.t. an isotropic log-concave distribution $\mathcal{D}$, $\eta' \le \eta/\epsilon$, and $\eta \le \Omega(\epsilon^3/\log(n/\epsilon))$. Then w.h.p. the hypothesis h returned by A_{mlcw} has advantage $\Omega\left(\frac{\epsilon^2}{n \log(n/\epsilon)}\right)$.*

4 Conclusion

There are relatively few algorithms for learning interesting classes of functions in the presence of malicious noise. We hope that our results will help lead to the development of more efficient algorithms for this challenging noise model.

As a concrete challenge for future work, we pose the following question: do there exist computationally efficient algorithms for learning halfspaces under *arbitrary* distributions in the presence of malicious noise? As of now no better results are known for this problem than the generic conversions of [12], which can be applied to any concept class. We feel that even a small improvement in the malicious noise rate that can be handled for halfspaces would be a very interesting result.

References

[1] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[3] N. H. Bshouty, Y. Li, and P. M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *J. Comp. Sys. Sci.*, 2009. To appear.

[4] J. Dunagan and S. Vempala. Optimal outlier removal in high-dimensional spaces. *J. Computer & System Sciences*, 68(2):335–373, 2004.

[5] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc. FOCS*, pages 563–576, 2006.

[6] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[7] Dmitry Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 4:101–117, 2003.

[8] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proc. FOCS*, pages 543–552. IEEE Computer Society, 2006.

[9] D. Haussler, M. Kearns, N. Littlestone, and M. Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95(2):129–161, 1991.

[10] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002.

[11] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[12] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

[13] A. Klivans and A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *Proc. FOCS*, pages 553–562, 2006.

[14] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

[15] N. Littlestone. Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proc. COLT*, pages 147–156, 1991.

[16] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures and Algorithms*, 30(3):307–358, 2007.

[17] W. Maass and G. Turan. How fast can a threshold gate learn? In *Computational Learning Theory and Natural Learning Systems: Volume I: Constraints and Prospects*, pages 381–414. MIT Press, 1994.

[18] Y. Mansour and M. Parnas. Learning conjunctions with noise under product distributions. *Information Processing Letters*, 68(4):189–196, 1998.

[19] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[20] D. Pollard. *Convergence of Stochastic Processes*. Springer Verlag, 1984.

[21] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[22] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[23] R. Servedio. PAC analogues of Perceptron and Winnow via boosting the margin. In *Proc. COLT*, pages 148–157, 2000.

[24] R. Servedio. *Efficient Algorithms in Computational Learning Theory*. PhD thesis, Harvard University, 2001.

[25] R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.

[26] J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines*. Cambridge University Press, 2000.

[27] L. Valiant. Learning disjunctions of conjunctions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 560–566, 1985.

A Definitions and Preliminaries

A.1 Learning with Malicious Noise

Given a probability distribution $\mathcal{D}$ over $\mathbf{R}^n$, and a target function $f : \mathbf{R}^n \to \{-1, 1\}$, we define the oracle $\mathrm{EX}_\eta(f, \mathcal{D})$ as follows:

- with probability $1 - \eta$ the oracle draws $\mathbf{x}$ according to $\mathcal{D}$, and outputs $(\mathbf{x}, f(\mathbf{x}))$, and
- with probability η the oracle outputs an arbitrary $(\mathbf{x}, y)$ pair. This "noisy" example can be thought of as being generated adversarially and can depend on the state of the learning algorithm and previous draws from the oracle.

Given a data set drawn from $\mathrm{EX}_\eta(f, \mathcal{D})$, we often refer to the examples $(\mathbf{x}, f(\mathbf{x}))$ (that came from $\mathcal{D}$) as "clean" examples and the remaining examples $(\mathbf{x}, y)$ as "dirty" examples.

For a set $\mathcal{S}$ of probability distributions and a set F of possible target functions, we say the a learning algorithm A learns F to accuracy $1 - \epsilon$ with respect to $\mathcal{S}$ in the presence of malicious noise at a rate η if the following holds: for any $f \in F$, and $\mathcal{D} \in \mathcal{S}$, given access to $\mathrm{EX}_\eta(f, \mathcal{D})$, with probability at least $1/2$, the output hypothesis h generated by A satisfies $\Pr_{\mathbf{x}\sim\mathcal{D}}[h(\mathbf{x}) \neq f(\mathbf{x})] \leq \epsilon$. (The probability of success may be amplified arbitrarily close to 1 using standard techniques [9].)

We note that for learning under the uniform distribution on the unit ball $\mathbb{S}^{n-1}$, we may assume w.l.o.g. that even noisy examples $(\mathbf{x}, y)$ have $\mathbf{x} \in \mathbb{S}^{n-1}$ – this is simply because a learning algorithm can trivially identify and ignore any noisy example $(\mathbf{x}, y)$ that has $\|\mathbf{x}\| \neq 1$.

A.2 Log-concave distributions

A probability distribution over $\mathbf{R}^n$ is said to be *log-concave* if its density function is $\exp(-\psi(\mathbf{x}))$ for a convex function ψ.

A probability distribution over $\mathbf{R}^n$is *isotropic* if the mean of the distribution is 0 and the covariance matrix is the identity, i.e. $\mathbf{E}[x_i x_j] = 1$ for $i = j$ and 0 otherwise.

Isotropic log-concave (henceforth abbreviated i.l.c.) distributions are a fairly broad class of distributions. It is well known that any distribution induced by taking a uniform distribution over an arbitrary convex set and applying a suitable linear transformation to make it isotropic is then isotropic and log-concave. For an excellent treatment on basic properties of log-concave distributions, see Lovasz and Vempala [16].

We will use the following facts:

Lemma 9 ([16]). *Let $\mathcal{D}$ be an isotropic log-concave distribution over $\mathbf{R}^n$ and $\mathbf{a} \in \mathbb{S}^{n-1}$ any direction. Then for $\mathbf{x}$ drawn according to $\mathcal{D}$, the distribution of $\mathbf{a} \cdot \mathbf{x}$ is an isotropic log-concave distribution over $\mathbf{R}$.*

Lemma 10 ([16]). *Any isotropic log-concave distribution $\mathcal{D}$ over $\mathbf{R}^n$ has light tails,*

$$\Pr_{\mathbf{x}\sim\mathcal{D}}[||\mathbf{x}|| > \beta\sqrt{n}] \leq e^{-\beta+1}.$$

If $n = 1$, the density of $\mathcal{D}$ is bounded:

$$\Pr_{x\sim\mathcal{D}}[\mathbf{x} \in [a, b]] \leq |b - a|.$$

B Proof of Lemma 8

Recall Lemma 8:

Lemma 8. *Suppose Algorithm A_{mlcw} is run using $\mathrm{EX}_{\eta'}(f, \mathcal{D}')$ where f is an origin-centered halfspace, $\mathcal{D}'$ is $(1/\epsilon)$-smooth w.r.t. an isotropic log-concave distribution $\mathcal{D}$, $\eta' \leq \eta/\epsilon$, and $\eta \leq \Omega(\epsilon^3/\log(n/\epsilon))$. Then w.h.p. the hypothesis h returned by A_{mlcw} has advantage $\Omega\left(\frac{\epsilon^2}{n\log(n/\epsilon)}\right)$.*

Before proving Lemma 8, we need to prove some uniformity results on non-noisy examples drawn from an isotropic, log-concave distribution. This will enable us to use outlier removal and averaging to find a weak learner.

B.1 Lemmas in support of Lemma 8

In this section, let us consider a single call to the weak learner with an oracle $EX_{\eta'}(f, \mathcal{D}')$ where $\mathcal{D}'$ is $(1/\epsilon)$-smooth with respect to an isotropic log-concave distribution $\mathcal{D}$ and $\eta' \leq \eta/\epsilon$. Our analysis will follow the same basic steps as Section 2.

A preliminary observation is that w.h.p. all clean examples drawn in Step 1 of Algorithm A_{mlcw} have $\|\mathbf{x}\| \leq \sqrt{3n\log m}$; indeed, for any given draw of $\mathbf{x}$ from $\mathcal{D}'$, the probability that $\|\mathbf{x}\| > \sqrt{3n\log m}$ is at most $\frac{e}{\epsilon m^3}$ by Lemma 10. Therefore, w.h.p., only noisy examples are removed in Step 2 of the algorithm, and we shall assume that the distributions $\mathcal{D}$ and $\mathcal{D}'$ are in fact supported entirely on $\{\mathbf{x} : \|\mathbf{x}\| \leq \sqrt{3n\log m}\}$. This assumption affects us in two ways: first, it costs us an additional $\frac{e}{\epsilon m^2}$ in the failure probability analysis below (which is not a problem and is in fact swallowed up by our "w.h.p." notation). Second, it means that the overall $1-\epsilon$ accuracy bound we establish for the entire learning algorithm may be slightly worse than the true value. This is because our final hypothesis may always be wrong on the examples $\mathbf{x}$ that have $\|\mathbf{x}\| > \sqrt{3n\log m}$ and are ignored in our analysis; however such examples have probability mass at most $\frac{e}{m^3}$ under the isotropic log-concave distribution $\mathcal{D}$ (again by Lemma 10), and thus the additional accuracy cost is at most $\frac{e}{m^3}$. Since $\epsilon \gg \frac{e}{m^3}$, this does not affect the overall correctness of our analysis. Note that a consequence of this assumption is that we can just take $h(\mathbf{x}) = \frac{\mathbf{v}\cdot\mathbf{x}}{3n\log m}$.

The remarks about high-probability statements and failure probabilities from Section 2.1 apply here as well, and as in Section 2 we write "w.h.p." as shorthand for "with probability $1 - 1/\mathrm{poly}(n/\epsilon)$."

We first show that the expected variance of $\mathcal{D}'$ in every direction is not too large:

Lemma 11. *For any $\mathbf{a} \in \mathbb{S}^{n-1}$ we have $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[(\mathbf{a}\cdot\mathbf{x})^2] = O(\log(1/\epsilon))$.*

Proof. For $\mathbf{x}$ chosen according to $\mathcal{D}$, the distribution of $\mathbf{a} \cdot \mathbf{x}$ is a unit variance log-concave distribution by Lemma 9. Thus, for any positive integer k,

$$\begin{aligned}
\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[(\mathbf{a}\cdot\mathbf{x})^2] &\le k + \sum_{i=k}^{\infty}(i+1)\Pr_{\mathbf{x}\sim\mathcal{D}'}[\mathbf{a}\cdot\mathbf{x}\in(i,i+1]] \\
&\le k + \sum_{i=k}^{\infty}(i+1)(1/\epsilon)\Pr_{\mathbf{x}\sim\mathcal{D}}[\mathbf{a}\cdot\mathbf{x}\in(i,i+1]] \\
&\le k + (1/\epsilon)\sum_{i=k}^{\infty}(i+1)\Pr_{\mathbf{x}\sim\mathcal{D}}[\mathbf{a}\cdot\mathbf{x}>i] \\
&\le k + (1/\epsilon)\sum_{i=k}^{\infty}(i+1)e^{-i+1} \le k + (1/\epsilon)\cdot\Theta(ke^{-k})
\end{aligned}$$

where the first inequality in the last line uses Lemmas 9 and 10. Setting $k = \Theta(\log(1/\epsilon))$ completes the proof. □

The following anticoncentration bound will be useful for proving that clean examples drawn from $\mathcal{D}'$ tend to be classified correctly with a large margin.

Lemma 12. *Let* $\mathbf{u} \in \mathbb{S}^{n-1}$. *Then*

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{u}\cdot\mathbf{x}|] \ge \epsilon/8.$$

Proof. Clearly

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{u}\cdot\mathbf{x}|] \ge (\epsilon/4)\Pr_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{u}\cdot\mathbf{x}| > \epsilon/4].$$

But by Lemma 10,

$$\Pr_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{u}\cdot\mathbf{x}| \le \epsilon/4] \;\le\; \frac{1}{\epsilon}\Pr_{\mathbf{x}\sim\mathcal{D}}[|\mathbf{u}\cdot\mathbf{x}| \le \epsilon/4] \;\le\; \frac{\epsilon/2}{\epsilon} = 1/2.$$

□

The next two lemmas are isotropic log-concave analogues of the uniform distribution Lemmas 2 and 3 respectively. The first one says that w.h.p. no direction a has much more variance than the expected variance in any direction:

Lemma 13. *W.h.p. over a random draw of* ℓ *clean examples* S_{clean} *from* $\mathcal{D}'$, *we have*

$$\max_{\mathbf{a}\in\mathbb{S}^{n-1}}\left\{\frac{1}{\ell}\sum_{(\mathbf{x},y)\in S_{\text{clean}}}(\mathbf{a}\cdot\mathbf{x})^2\right\} \le O(1)\left(\log\frac{1}{\epsilon} + \frac{n^{3/2}\log^2 m}{\sqrt{\ell}}\right).$$

Proof. By Lemma 11, for any $\mathbf{a} \in \mathbb{S}^{n-1}$ we have

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[(\mathbf{a}\cdot\mathbf{x})^2] = \Theta(\log(1/\epsilon)).$$

Since as remarked earlier we may assume $\mathcal{D}'$ is supported on $\{\mathbf{x} : \|\mathbf{x}\| \le \sqrt{3n\log m}\}$, we may apply Lemmas 19 and 20 (see Appendix D) with functions $f_{\mathbf{a}}$ defined by $f_{\mathbf{a}} = \frac{(\mathbf{a}\cdot\mathbf{x})^2}{3n\log m}$ This completes the proof. □

The second lemma says that for a sufficiently large clean data set, w.h.p. no direction has too many examples lying too far out in that direction:

Lemma 14. *For any $\beta > 0$ and $\kappa > 1$, if S_{clean} is a set of $\ell \geq \frac{O(1)\epsilon e^{\beta}\left(n \ln\left(\epsilon e^{\beta}\right)+\log m\right)}{(1+\kappa)\ln(1+\kappa)}$ clean examples drawn from $\mathcal{D}'$, then w.h.p. we have*

$$\max_{\mathbf{a}\in\mathbb{S}^{n-1}} \left\{ \frac{1}{\ell} \sum_{x \in S_{\text{clean}}} \mathbf{1}_{(a\cdot x)^2 > \beta^2} \right\} \leq (1+\kappa)\left(\frac{1}{\epsilon}\right) e^{-\beta+1}.$$

Proof. Lemma 10 implies that for the original isotropic log-concave distribution $\mathcal{D}$, we have $\Pr_{\mathbf{x}\sim\mathcal{D}}[(\mathbf{a}\cdot\mathbf{x})^2 > \beta] \leq e^{-\beta+1}$. Since $\mathcal{D}'$ is $(1/\epsilon)$-smooth with respect to $\mathcal{D}$, this implies that

$$\Pr_{\mathbf{x}\in\mathcal{D}'}[(\mathbf{a}\cdot\mathbf{x})^2 > \beta] \leq \frac{e^{-\beta+1}}{\epsilon}. \tag{6}$$

In the proof of Lemma 3, we observed that the VC-dimension of

$$\{\mathbf{1}_{(\mathbf{a}\cdot\mathbf{x})^2>\beta} : \mathbf{a} \in \mathbf{R}^n, \beta \in \mathbf{R}\}$$

is $O(n)$, so applying Lemma 21 with (6) completes the proof of this lemma. □

The following is an isotropic log-concave analogue of Corollary 1, establishing that not too many clean examples are removed in the outlier removal step:

Corollary 2. *W.h.p. over the random draw of the m-element sample S from $EX_{\eta'}(f, \mathcal{D}')$, the number of clean examples removed during any execution of the outlier removal step (final substep of Step 2) in Algorithm A_{mlcw} is at most $6m\epsilon^3/n^4$.*

Proof. Since the true noise rate η is assumed sufficiently small, the value $\eta' \leq \eta/\epsilon$ is at most $\epsilon/4$, and thus w.h.p. the number ℓ of clean examples in S is at least (say) $m/2$. We would like to apply Lemma 14 with $\kappa = (n/\epsilon)^{c_0-4}$ and $\beta = c_0 \log(n/\epsilon)$, and we may do this since we have

$$\frac{O(1)\epsilon e^{\beta}\left(n \ln\left(\epsilon e^{\beta}\right) + \log m\right)}{(1+\kappa)\ln(1+\kappa)} \leq \frac{O(1)\epsilon(n/\epsilon)^{c_0} n \log m}{(n/\epsilon)^{c_0-4}\log m} \leq O(1)n^5/\epsilon^3 \ll \frac{m}{2} \leq \ell$$

for a suitable fixed $\text{poly}(n/\epsilon)$ choice of m. Since clean points are only removed if they have $(\mathbf{a}\cdot\mathbf{x})^2 \geq \beta^2$, Lemma 14 gives us that the number of clean points removed is at most

$$m(1+\kappa)\cdot\frac{1}{\epsilon}e^{-\beta+1} \leq m\frac{(6/\epsilon)(n/\epsilon)^{c_0-4}}{(n/\epsilon)^{c_0}} \leq 6m\epsilon^3/n^4.$$

□

Not surprisingly, the following lemma is an analogue of Lemma 4; it lower bounds the number of dirty examples that are removed in the outlier removal step.

Lemma 15. *W.h.p. over the random draw of S, any time Algorithm A_{mlcw} executes the outlier removal step it removes at least $\frac{m}{3n}$ noisy examples.*

Proof. Since our ultimate goal is only to prove that the algorithm succeeds for some η which is $o(\epsilon)$, we may assume without loss of generality that the original noise rate η is less than $\epsilon/4$. This means that $\eta' < 1/4$, and consequently a Chernoff bound gives that w.h.p. the fraction $\widehat{\eta}'$ of noisy examples in S at at the beginning of the weak learner's training is at most $1/2$. And Lemma 13 implies that for a sufficiently large polynomial choice of m, we have that w.h.p. for all $a \in \mathbb{S}^{n-1}$, the following holds for all the clean examples in the data before any examples were removed:

$$\sum_{(\mathbf{x},y)\in S_{\text{clean}}} (\mathbf{a}\cdot\mathbf{x})^2 \le cm\log(1/\epsilon) \tag{7}$$

where c is an absolute constant. We say that a random sample that meets all these requirements is "reasonable." We now set the constant c_0 that is used in the specification of A_{mlc} to be $2c+2$. We will now show that, for any reasonable sample S, the number of noisy examples removed during the first execution of the outlier removal step of A_{mu} is at least $\frac{m}{3n}$.

If we remove examples using direction $\mathbf{w}$ then it means $\sum_{x\in S}(\mathbf{w}\cdot\mathbf{x})^2 \ge c_0 m\log(n/\epsilon)$. Since S is reasonable, by (7) the contribution to the sum from the clean examples that have survived until this point is at most $cm\log(1/\epsilon)$ so we must have

$$\sum_{(\mathbf{x},y)\in S_{\text{dirty}}} (\mathbf{w}\cdot\mathbf{x})^2 \ge (c_0 - c)m\log(n/\epsilon).$$

Let $S_{\text{dirty}} = N \cup F$ where N is the examples $(\mathbf{x},y)$ for which $\mathbf{x}$ satisfies $(\mathbf{w}\cdot\mathbf{x})^2 \le c_0\log(n/\epsilon)$ and F is the other points. We have

$$\sum_{(\mathbf{x},y)\in N} (\mathbf{w}\cdot\mathbf{x})^2 \le c_0\widehat{\eta}' m\log(n/\epsilon).$$

and so, since $||\mathbf{x}|| \le \sqrt{3n\log m}$ implies that $(\mathbf{w}\cdot\mathbf{x})^2 \le 3n\log m$ for all unit length $\mathbf{w}$, we have

$$\begin{aligned}
|F| \ge \sum_{(\mathbf{x},y)\in F} \frac{(\mathbf{w}\cdot\mathbf{x})^2}{3n\log m} &= \sum_{(\mathbf{x},y)\in S_{\text{dirty}}} \frac{(\mathbf{w}\cdot\mathbf{x})^2}{3n\log m} - \sum_{(\mathbf{x},y)\in N} \frac{(\mathbf{w}\cdot\mathbf{x})^2}{3n\log m} \\
&\ge \frac{(c_0-c)m\log(n/\epsilon) - c_0\widehat{\eta}' m\log(n/\epsilon)}{3n\log m} \\
&\ge \frac{m\log(n/\epsilon)}{3n\log m} \ge \frac{m}{3n},
\end{aligned}$$

where the next-to-last inequality uses $\eta' \le 1/2$ and $c_0 = 2(c+1)$, and the final one uses $m \ge n/\epsilon$. The points in F are precisely the ones that are removed, and thus the lemma is proved. □

B.2 Proof of Lemma 8

We first note that w.h.p. the weak learner must terminate after at most $3n$ iterations of outlier removal.

Let $\mathbf{u}$ be the unit length normal vector of the separating halfspace for the target function f. The advantage of h with respect to $\mathcal{D}'$ can be expressed as

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[h(\mathbf{x})f(\mathbf{x})] = \frac{\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[(\mathbf{v}\cdot\mathbf{x})f(\mathbf{x})]}{3n\log m} \tag{8}$$

and so we shall work on lower bounding $\mathbf{E}_{\mathbf{x}\in\mathcal{D}'}[(\mathbf{v}\cdot\mathbf{x})f(\mathbf{x})]$.

As in the proof of Lemma 6, let

- S_{clean} be all of the clean examples in the initial sample S, and S'_{clean} be those that are not removed in any stage of outlier removal;
- S_{dirty} be all of the dirty examples in the initial sample S, and S'_{dirty} be those that are not removed in any stage of outlier removal;
- $\eta' = \frac{|S'_{\text{dirty}}|}{|S'_{\text{clean}} \cup S'_{\text{dirty}}|}$, i.e. the noise rate among the examples that survive until the end of the training of the weak learner, and
- $\alpha = \frac{|S_{\text{clean}} - S'_{\text{clean}}|}{|S'_{\text{clean}} \cup S'_{\text{dirty}}|}$, the ratio of the number of clean points that were erroneously removed to the size of the final surviving data set.

As before we write S' for $S'_{\text{clean}} \cup S'_{\text{dirty}}$. Also as before, let $\mathbf{v}_{\text{clean}}$ be the average of *all* the clean examples, $\mathbf{v}'_{\text{dirty}}$ be the average of the dirty (noisy) examples that were not deleted, and $\mathbf{v}_{\text{del}}$ be the average of the clean examples that were deleted. Then arguing exactly as before, we have

$$\mathbf{v} = (1 - \eta' + \alpha)\mathbf{v}_{\text{clean}} + \eta'\mathbf{v}'_{\text{dirty}} - \alpha\mathbf{v}_{\text{del}}. \tag{9}$$

The expectation of $\mathbf{v}_{\text{clean}}$ will play a special role in the analysis:

$$\mathbf{v}^*_{\text{clean}} \stackrel{def}{=} \mathbf{E}_{\mathbf{x}\in\mathcal{D}'}[f(\mathbf{x})\mathbf{x}].$$

Once again, we will demonstrate the limited effect of $\mathbf{v}'_{\text{dirty}}$ by bounding its length. This time, the outlier removal enforces the fact that, for any $\mathbf{w} \in \mathbb{S}^{n-1}$, we have

$$\sum_{(\mathbf{x},y)\in S} (\mathbf{w}\cdot\mathbf{x})^2 \le c_0 m \log(n/\epsilon).$$

Applying this for the unit vector $\mathbf{w}$ in the direction of $\mathbf{v}'_{\text{dirty}}$ as was done in Lemma 6, this implies

$$\|\mathbf{v}'_{\text{dirty}}\| \le \sqrt{\frac{c_0 m \log(n/\epsilon)}{|S'_{\text{dirty}}|}}.$$

Next, let us apply this to bound an expression that captures the average harm done by $\mathbf{v}'_{\text{dirty}}$.

$$\begin{aligned} |\mathbf{E}_{\mathbf{x}\in\mathcal{D}'}[f(\mathbf{x})(\mathbf{v}'_{\text{dirty}}\cdot\mathbf{x})]| &= |\mathbf{v}'_{\text{dirty}}\cdot\mathbf{v}^*_{\text{clean}}| \\ &\le \sqrt{\frac{c_0 m \log(n/\epsilon)}{|S'_{\text{dirty}}|}}\,||\mathbf{v}^*_{\text{clean}}||. \end{aligned} \tag{10}$$

To show that $\mathbf{v}_{\text{clean}}$ plays a relatively large role, it is helpful to lower bound the length of $\mathbf{v}^*_{\text{clean}}$. We do this by lower bounding the length of its projection onto the unit normal vector $\mathbf{u}$ of the target as follows:

$$\mathbf{v}^*_{\text{clean}}\cdot\mathbf{u} = \mathbf{E}[(f(\mathbf{x})\mathbf{x})\cdot\mathbf{u}] = \mathbf{E}[\text{sgn}(\mathbf{u}\cdot\mathbf{x})(\mathbf{x}\cdot\mathbf{u})] = \mathbf{E}[|\mathbf{x}\cdot\mathbf{u}|] \ge \epsilon/8,$$

by Lemma 12. Since $\mathbf{u}$ is unit length, this implies

$$||\mathbf{v}^*_{\text{clean}}|| \geq \epsilon/8. \tag{11}$$

Armed with this bound, we can now lower bound the benefit imparted by $\mathbf{v}_{\text{clean}}$:

$$\begin{aligned}\mathbf{E}_{\mathbf{z}\in\mathcal{D}'}[f(\mathbf{z})(\mathbf{v}_{\text{clean}}\cdot\mathbf{z})] &= \frac{1}{S_{\text{clean}}}\sum_{(\mathbf{x},y)\in S_{\text{clean}}}\mathbf{E}_{\mathbf{z}\in\mathcal{D}'}[yf(\mathbf{z})(\mathbf{x}\cdot\mathbf{z})]\\ &= \frac{1}{S_{\text{clean}}}\sum_{(\mathbf{x},y)\in S_{\text{clean}}}(y\mathbf{x})\cdot\mathbf{v}^*_{\text{clean}}.\end{aligned}$$

Since $\mathbf{E}[(y\mathbf{x})\cdot\mathbf{v}^*_{\text{clean}}] = ||\mathbf{v}^*_{\text{clean}}||^2$, and $(y\mathbf{x})\cdot\mathbf{v}^*_{\text{clean}} \in [-3n\log m, 3n\log m]$, a Hoeffding bound implies that w.h.p.

$$\mathbf{E}_{\mathbf{z}\in\mathcal{D}'}[f(\mathbf{z})(\mathbf{v}_{\text{clean}}\cdot\mathbf{z})] \geq ||\mathbf{v}^*_{\text{clean}}||^2 - O(n\log^{3/2} m)/\sqrt{|S_{\text{clean}}|}.$$

Since the noise rate η' is at most η/ϵ and η certainly less than $\epsilon/4$ as discussed above, another Hoeffding bound gives that w.h.p. $|S_{\text{clean}}|$ is at least $m/2$; thus for a suitably large polynomial choice of m, using (11) we have

$$\mathbf{E}_{\mathbf{z}\in\mathcal{D}'}[f(\mathbf{z})(\mathbf{v}_{\text{clean}}\cdot\mathbf{z})] \geq ||\mathbf{v}^*_{\text{clean}}||^2 - O(n\log^{3/2} m)/\sqrt{m/2} \geq \frac{||\mathbf{v}^*_{\text{clean}}||^2}{2}. \tag{12}$$

Now we are ready to put our bounds together and lower bound the advantage of $\mathbf{v}$. We have

$$\begin{aligned}\mathbf{E}_{\mathbf{x}\in\mathcal{D}'}[f(\mathbf{x})(\mathbf{v}\cdot\mathbf{x})] = (1-\eta'+\alpha)&\mathbf{E}[f(\mathbf{x})(\mathbf{v}_{\text{clean}}\cdot\mathbf{x})]\\ &+\eta'\mathbf{E}[f(\mathbf{x})(\mathbf{v}'_{\text{dirty}}\cdot\mathbf{x})] - \alpha\mathbf{E}[f(\mathbf{x})(\mathbf{v}_{\text{del}}\cdot\mathbf{x})].\end{aligned}$$

We bound each of the three contributions in turn. First, using $1-\eta' \geq 1/2$ and (12), we have $(1-\eta'+\alpha)\mathbf{E}[f(\mathbf{x})(\mathbf{v}_{\text{clean}}\cdot\mathbf{x})] \geq \frac{||\mathbf{v}^*_{\text{clean}}||^2}{4}$.

Next, by (10), arguing as we did before equation (5), we have $|\eta'\mathbf{E}[f(\mathbf{x})(\mathbf{v}'_{\text{dirty}}\cdot\mathbf{x})]| \leq \sqrt{c_0\eta'\log(n/\epsilon)}||\mathbf{v}^*_{\text{clean}}||$. Since we may assume that $\eta \leq c'\epsilon^3/\log(n/\epsilon)$ for as small a fixed constant c' as we like (recall the overall bound of Theorem 2), we get

$$\sqrt{c_0\eta'\log(n/\epsilon)}||\mathbf{v}^*_{\text{clean}}|| \leq (\epsilon/64)||\mathbf{v}^*_{\text{clean}}||$$

(for a suitably small constant choice of c'), and this is less than $\frac{||\mathbf{v}^*_{\text{clean}}||^2}{8}$ since $||\mathbf{v}^*_{\text{clean}}|| \geq \epsilon/8$.

Finally Corollary 2, together with the fact that there are at most $3n$ iterations of outlier removal and the final surviving data set is of size at least $m/4$, gives us that $\alpha \leq \frac{(3n)(6m\epsilon^3/n^4)}{m/4}$, which (recalling that both $\mathbf{v}_{\text{del}}$ and all $\mathbf{x}$ in the support of $\mathcal{D}'$ have norm at most $\sqrt{3n\log m}$) means that $|\alpha\mathbf{E}[f(\mathbf{x})(\mathbf{v}_{\text{del}}\cdot\mathbf{x})| = o(\epsilon^2)$.

Combining all these bounds, we get

$$\mathbf{E}_{\mathbf{x}\in\mathcal{D}'}[f(\mathbf{x})(\mathbf{v}\cdot\mathbf{x})] \geq \frac{||\mathbf{v}^*_{\text{clean}}||^2}{4} - \frac{||\mathbf{v}^*_{\text{clean}}||^2}{8} - o(\epsilon^2) \geq \frac{\epsilon^2}{1024}$$

by (11). Together with (8), the proof of Lemma 8 is completed.

C Learning under isotropic log-concave distributions with adversarial label noise

C.1 The Model

We now define the model of learning with adversarial label noise under isotropic log-concave distributions. In this model the learning algorithm has access to an oracle that provides independent random examples drawn according to a fixed distribution P on $\mathbf{R}^n \times \{-1, 1\}$, where

- the marginal distribution over $\mathbf{R}^n$ is isotropic log-concave, and
- there is a halfspace f such that $\Pr_{(\mathbf{x},y)\sim P}[f(\mathbf{x}) \neq y] = \eta$.

The parameter η is the *noise rate*. As usual, the goal of the learner is to output a hypothesis h such that $\Pr_{(\mathbf{x},y)\sim \mathcal{D}}[h(\mathbf{x}) \neq y] \leq \epsilon$; if an algorithm achieves this goal, we say it learns to accuracy $1 - \epsilon$ in the presence of adversarial label noise at rate η.

C.2 The Algorithm

Like the algorithm A_{mlc} considered in the last section, the algorithm A_{alc} studied in this section applies the smooth boosting algorithm of Lemma 7 to a weak learner that performs averaging. The weak learner A_{alcw} behaves as follows:

1. Draw a set S of m examples according to P' (the oracle for a modified distribution provided by the boosting algorithm).
2. Remove all examples $(\mathbf{x}, y)$ such that $||\mathbf{x}|| > \sqrt{3n \log m}$ from S.
3. Let $\mathbf{v} = \frac{1}{|S|} \sum_{(\mathbf{x},y)\in S} y\mathbf{x}$. Return the confidence-rated classifier h defined by $h(\mathbf{x}) = \frac{\mathbf{v}\cdot\mathbf{x}}{3n \log m}$, if $|\mathbf{v} \cdot \mathbf{x}| \leq 3n \log m$, and $h(\mathbf{x}) = \mathrm{sgn}(\mathbf{v} \cdot \mathbf{x})$ otherwise.

C.3 Claim about the weak learner

As in the previous section, the heart of our analysis will be to analyze the weak learner. We omit discussing the application of the smooth boosting algorithm here, as it is nearly identical to Section 3.

Lemma 16. *Let P' be a distribution that is $(1/\epsilon)$-smooth with respect to a joint distribution on $\mathbf{R}^n \times \{-1, 1\}$ whose marginal on $\mathbf{R}^n$ is isotropic and log-concave. Further, assume there exists a linear threshold function f such that $\Pr_{(\mathbf{x},y)\sim P'}[f(\mathbf{x}) \neq y] \leq \eta/\epsilon$ and $\eta \leq \Omega(\frac{\epsilon^3}{\log(1/\epsilon)})$. Then with high probability, A_{alcw} outputs a hypothesis with advantage $\Omega(\frac{\epsilon^2}{n \log(n/\epsilon)})$.*

C.4 Lemmas in support of Lemma 16

In this section, let us focus our attention on a single call to the weak learner. Let P' be the distribution provided to the weak learner, and let $\mathcal{D}'$ be the marginal on $\mathbf{R}^n$. As in Section 3, we may assume that the support of $\mathcal{D}'$ lies entirely on $\mathbf{x}$ such that $||\mathbf{x}|| \leq \sqrt{3n \log m}$ (this negligibly affects the final bounds obtained in our analyses).

By Lemma 7 we immediately have

Lemma 17 **([25]).** *P' is $(1/\epsilon)$-smooth with respect to P.*

The following technical lemma will be used to limit the ability of an adversary for choosing P' to concentrate a lot of noise in one direction.

Lemma 18. *Let E be any event with positive probability under $\mathcal{D}'$, and let $\kappa = \mathcal{D}'(E)$. For any unit length $\mathbf{a} \in \mathbf{R}^n$, $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E] = O\left(\log \frac{1}{\kappa\epsilon}\right)$.*

Proof. Let β be such that $\Pr_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| > \beta] = \kappa$. By Lemmas 9 and 10, together with the fact that $\mathcal{D}'$ is $(1/\epsilon)$ smooth with respect to $\mathcal{D}$, we have

$$\kappa \le \frac{1}{\epsilon} e^{-\beta+1}$$

which implies $\beta \le 1 + \log\left(\frac{2}{\epsilon\kappa}\right)$.

Let F be the event that $|\mathbf{a}\cdot\mathbf{x}| > \beta$. We will show that $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E] \le \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F]$, and then bound $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F]$. If $\Pr[(E-F)\cup(F-E)] = 0$, then, obviously, $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E] = \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F]$. Suppose $\Pr[(E-F)\cup(F-E)] > 0$. Then

$$\begin{aligned}
&\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E]\\
&= \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E\cap F]\Pr[E\cap F] + \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E-F]\Pr[E-F]\\
&= \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E\cap F]\Pr[E\cap F] + \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E-F]\Pr[F-E]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\text{(because } \Pr[E] = \Pr[F]\text{)}\\
&< \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E\cap F]\Pr[E\cap F] + \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F-E]\Pr[F-E],
\end{aligned}$$

because for every $\mathbf{x} \in E-F$ and every $\mathbf{x}' \in F-E$,

$$|\mathbf{a}\cdot\mathbf{x}| \le \beta < |\mathbf{a}\cdot\mathbf{x}'|.$$

But

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E\cap F]\Pr[E\cap F] + \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F-E]\Pr[F-E] = \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F],$$

so

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid E] < \mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F]. \tag{13}$$

Now, setting $b = \lfloor\beta\rfloor$, we have

$$\begin{aligned}
\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \mid F] &\le \frac{1}{\mathcal{D}'(F)} \sum_{i=b} (i+1) \Pr_{\mathbf{x}\sim\mathcal{D}'}[|\mathbf{a}\cdot\mathbf{x}| \in (i, i+1]]\\
&\le \frac{1}{\mathcal{D}'(F)} \sum_{i=b} (i+1) e^{-i+1}\\
&= \frac{1}{\mathcal{D}'(F)} \left(O(\frac{e^{-b}b}{\epsilon})\right)\\
&= O(b),
\end{aligned}$$

since $\mathcal{D}'(F) = \Theta(e^{-b}/\epsilon)$. Combining with (13) completes the proof. □

C.5 Proof of Lemma 16

Fix some halfspace f such that $\Pr_{(\mathbf{x},y)\sim P}[f(\mathbf{x}) \neq y] = \eta$, and let $\mathbf{u}$ be the unit normal vector of its separating hyperplane.

Let P' be the joint distribution given to $A_{\rm alcw}$ and let $\mathcal{D}'$ be its marginal on $\mathbf{R}^n$. Lemma 17 implies that $\mathcal{D}'$ is $(1/\epsilon)$-smooth with respect to the original marginal distribution $\mathcal{D}$.

First, we bound the advantage with respect to P' in terms of the tendency of h to agree with the best linear function f:

$$\mathbf{E}_{(\mathbf{x},y)\sim P'}[h(\mathbf{x})y] \geq \mathbf{E}_{(\mathbf{x},y)\sim P'}[h(\mathbf{x})f(\mathbf{x})] - \eta = \mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[h(\mathbf{x})f(\mathbf{x})] - \eta. \tag{14}$$

Furthermore, we have

$$\mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[h(\mathbf{x})f(\mathbf{x})] = \mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}\left[\frac{f(\mathbf{x})(\mathbf{x}\cdot\mathbf{v})}{3n\log m}\right] \tag{15}$$

so we will work on bounding $\mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[f(\mathbf{x})(\mathbf{x}\cdot\mathbf{v})]$.

Let $P'_{\rm clean}$ be obtained by conditioning a random draw $(\mathbf{x},y)$ from P' on the event that $f(\mathbf{x}) = y$. Define $P'_{\rm dirty}$ analogously, and let $\mathcal{D}'_{\rm clean}$ and $\mathcal{D}'_{\rm dirty}$ be the corresponding marginals on $\mathbf{R}^n$. Let

$$\begin{aligned}
\mathbf{v}^*_{\rm clean} &= \mathbf{E}_{(\mathbf{x},y)\sim P'_{\rm clean}}[y\mathbf{x}]\\
\mathbf{v}^*_{\rm dirty} &= \mathbf{E}_{(\mathbf{x},y)\sim P'_{\rm dirty}}[y\mathbf{x}]\\
\mathbf{v}^*_{\rm correct} &= \mathbf{E}_{\mathbf{x}\in \mathcal{D}'}[f(\mathbf{x})\mathbf{x}].
\end{aligned}$$

Note that

$$\mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[f(\mathbf{x})(\mathbf{x}\cdot\mathbf{v})] = \mathbf{v}^*_{\rm correct}\cdot\mathbf{v} = \frac{1}{m}\sum_{(\mathbf{x},y)\in S}\mathbf{v}^*_{\rm correct}\cdot(y\mathbf{x}). \tag{16}$$

Equation (16) expresses $\mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[f(\mathbf{x})(\mathbf{x}\cdot\mathbf{v})]$, which is closely related to the advantage of h through (15) and (14), as a sum of independent random variables, one for each example. We will bound $\mathbf{E}_{\mathbf{x}\sim \mathcal{D}'}[f(\mathbf{x})(\mathbf{x}\cdot\mathbf{v})]$ by bounding the expected effect of a random example on its value, and applying a Hoeffding bound.

Let $\eta' = \Pr_{(\mathbf{x},y)\sim P'}[f(\mathbf{x}) \neq y]$. Since P' is $1/\epsilon$-smooth with respect to P, we have $\eta' \leq \eta/\epsilon$. We can rearrange the effect of a random example as follows

$$\begin{aligned}
\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (y\mathbf{x})] &= (1-\eta')\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y = f(\mathbf{x})] \\
&\qquad\qquad +\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (-f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&= (1-\eta')\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y = f(\mathbf{x})] \\
&\qquad\qquad +\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&\qquad\qquad -\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&\qquad\qquad +\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (-f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&= \mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})] \\
&\qquad\quad -2\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&= ||\mathbf{v}^*_{\mathrm{correct}}||^2 - 2\eta'\mathbf{E}_{(\mathbf{x},y)\sim P'}[\mathbf{v}^*_{\mathrm{correct}} \cdot (f(\mathbf{x})\mathbf{x})|y \neq f(\mathbf{x})] \\
&= ||\mathbf{v}^*_{\mathrm{correct}}||^2 - 2\eta'\mathbf{v}_{\mathrm{correct}} \cdot \mathbf{v}_{\mathrm{dirty}} \\
&\geq ||\mathbf{v}^*_{\mathrm{correct}}||^2 - 2\eta'||\mathbf{v}^*_{\mathrm{correct}}|| \cdot ||\mathbf{v}^*_{\mathrm{dirty}}|| \\
&\geq \frac{1}{2}||\mathbf{v}^*_{\mathrm{correct}}||^2 - 4(\eta')^2||\mathbf{v}^*_{\mathrm{dirty}}||^2, \qquad (17)
\end{aligned}$$

The last line follows from the fact that $q^2 - qr \geq (q^2 - r^2)/2$ for all real q, r.

So now our goals are a lower bound on $||\mathbf{v}^*_{\mathrm{correct}}||$ and an upper bound on $||\mathbf{v}^*_{\mathrm{dirty}}||$.

We can lower bound $||\mathbf{v}^*_{\mathrm{correct}}||$ essentially the same way we did before, by lower bounding its projection onto the “target” normal vector $\mathbf{u}$:

$$\mathbf{v}^*_{\mathrm{correct}} \cdot \mathbf{u} = \mathbf{E}[(f(\mathbf{x})\mathbf{x}) \cdot \mathbf{u}] = \mathbf{E}[\mathrm{sgn}(\mathbf{u} \cdot \mathbf{x})(\mathbf{x} \cdot \mathbf{u})] = \mathbf{E}[|\mathbf{x} \cdot \mathbf{u}|] \geq \epsilon/16, \qquad (18)$$

by Lemma 12.

We upper bound $||\mathbf{v}^*_{\mathrm{dirty}}||$ as follows:

$$\begin{aligned}
||\mathbf{v}^*_{\mathrm{dirty}}||^2 &= \mathbf{v}^*_{\mathrm{dirty}} \cdot E_{\mathbf{x}\in\mathcal{D}'_{\mathrm{dirty}}}[(1 - f(\mathbf{x}))\mathbf{x}] \\
&= ||\mathbf{v}^*_{\mathrm{dirty}}|| \cdot E_{\mathbf{x}\in\mathcal{D}'_{\mathrm{dirty}}}\left[\left(\frac{\mathbf{v}^*_{\mathrm{dirty}}}{||\mathbf{v}^*_{\mathrm{dirty}}||}\right) \cdot (1 - f(\mathbf{x}))\mathbf{x}\right] \\
&\leq ||\mathbf{v}^*_{\mathrm{dirty}}|| \cdot E_{\mathbf{x}\in\mathcal{D}'_{\mathrm{dirty}}}\left[\left|\left(\frac{\mathbf{v}^*_{\mathrm{dirty}}}{||\mathbf{v}^*_{\mathrm{dirty}}||}\right) \cdot \mathbf{x}\right|\right] \\
&\leq ||\mathbf{v}^*_{\mathrm{dirty}}||O(\log(1/(\eta'\epsilon)))
\end{aligned}$$

by Lemma 18. Thus $||\mathbf{v}^*_{\mathrm{dirty}}|| \leq O(\log(1/(\eta'\epsilon)))$.

Combining this with (18), we have that if

$$(\eta')^2 \cdot (\log(1/(\eta'\epsilon))^2 \leq c\epsilon^2 \qquad (19)$$

for a suitably small constant c, then (17) is at least $\Omega(\epsilon^2)$. Thus, for such η', by (16) we have that $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[f(\mathbf{x})(\mathbf{x} \cdot \mathbf{v})]$ is a sum of m i.i.d. random variables, each with mean at least $\Omega(\epsilon^2)$, and coming from an interval of length $O(n \log m)$. Applying the standard Hoeffding bound, polynomially many examples suffice for $\mathbf{E}_{\mathbf{x}\sim\mathcal{D}'}[f(\mathbf{x})(\mathbf{x} \cdot \mathbf{v})] \geq \Omega(\epsilon^2)$. Combining with (15) and (14), we are almost done: it remains only to observe that (19) holds as long as η is at most a small constant multiple of $\epsilon^2/\log(1/\epsilon)$ (recalling that $\eta' \leq \eta/\epsilon$).

D Proof of Lemma 2

Let us start with a couple of definitions and a bound from the literature.

Definition 1 (VC-dimension). *A set F of $\{-1,1\}$-valued functions defined on a common domain X* shatters *$x_1, ..., x_d$ if every sequence $y_1, ..., y_d \in \{-1,1\}$ of function values has a function f such that $f(x_1) = y_1, ..., f(x_d) = y_d$. The VC-dimension of F is the size of the largest set shattered by F.*

Definition 2 (pseudo-dimension). *For a set F of real-valued functions defined on a common domain X, the* pseudo-dimension *of F is the VC-dimension of $\{\mathrm{sign}(f(\cdot)-\theta) : f \in F, \theta \in \mathbf{R}\}$.*

Lemma 19 ([20]). *Let F be a set of real-valued functions defined on a common domain X taking values in $[0,1]$, and let d be the pseudo-dimension of F. Let $\mathcal{D}$ be a probability distribution over X. Then if $x_1, ..., x_m$ are obtained by drawing m times independently according to $\mathcal{D}$, for any $\delta > 0$,*

$$\Pr\left[\exists f \in F, \frac{1}{m}\sum_{s=1}^{m} f(x_s) > E_{\mathcal{D}}[f] + \sqrt{\frac{cd\log(1/\delta)}{m}}\right] \leq \delta,$$

where $c > 0$ is an absolute constant.

Now, let us bound the pseudo-dimension of the class of functions that we need.

Lemma 20. *Let F_n consist of the functions f from $\mathbf{R}^n$ to $\mathbf{R}$ which can be defined by $f(\mathbf{x}) = (\mathbf{a} \cdot \mathbf{x})^2$ for some $\mathbf{a} \in \mathbf{R}^n$. The pseudo-dimension of F_n is at most $O(n)$.*

Proof. According to the definition, the pseudo dimension of F_n is the VC-dimension of the set G_n of $\{-1,1\}$-valued functions $g_{\mathbf{a},\theta}$ defined by $g_{\mathbf{a},\theta}(\mathbf{x}) = \mathrm{sign}((\mathbf{a} \cdot \mathbf{x})^2 - \theta)$. Each $g_{\mathbf{a},\theta}$ is equivalent to an OR of two halfspaces:

$$\mathbf{a} \cdot \mathbf{x} \geq \sqrt{\theta} \quad \text{OR} \quad (-\mathbf{a}) \cdot \mathbf{x} \geq \sqrt{\theta}$$

Thus the VC-dimension of G_n is at most the VC-dimension of the class of all ORs of two halfspaces, which is known to be $O(n)$ (see [2]). $\square$

Applying Lemmas 19 and 20, we obtain Lemma 2.

E Proof of Lemma 3

We will use the following, which strengthens bounds like Lemma 19 when the expectations being estimated are small. It differs from most bounds of this type by providing an especially strong bound on the probability that the estimates are *much* larger than the true expectations.

Lemma 21 ([3]). *Suppose F is a set of $\{0,1\}$-valued functions with a common domain X. Let d be the VC-dimension of F. Let $\mathcal{D}$ be a probability distribution over X. Choose $\alpha > 0$ and $K \geq 4$. Then if*

$$m \geq \frac{c\left(d \log \frac{1}{\alpha} + \log \frac{1}{\delta}\right)}{\alpha K \log K},$$

where c is an absolute constant, then

$$\Pr_{\mathbf{u} \sim \mathcal{D}^m}[\exists f \in F,\ \mathbf{E}_{\mathcal{D}}(f) \leq \alpha \textit{ but } \hat{\mathbf{E}}_{\mathbf{u}}(f) > K\alpha] \leq \delta,$$

where $\hat{\mathbf{E}}_{\mathbf{u}}(f) = \frac{1}{m}\sum_{i=1}^{m} f(u_i)$.

To prove Lemma 3, we first use the fact that, for any fixed $\mathbf{a} \in \mathbb{S}^{n-1}$ and $\beta > 0$, it is known (see [11]) that

$$\Pr_{x \in \mathbb{S}^{n-1}}[|\mathbf{a} \cdot \mathbf{x}| > \beta] \leq e^{-\beta^2 n/2}.$$

Further, as in the proof of Lemma 2, we have that

$$|\mathbf{a} \cdot \mathbf{x}| > \beta \quad \text{if and only if} \quad \mathbf{a} \cdot \mathbf{x} > \beta \text{ OR } (-\mathbf{a}) \cdot \mathbf{x} > \beta,$$

so that the set of events whose probabilities we need to estimate is contained in the set of unions of pairs of halfspaces. The VC-dimension of the latter is known to be $O(n)$, so applying Lemma 21 completes the proof.