# 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 $\epsilon$:

- 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 $\eta = \Omega(\epsilon^3/\log(1/\epsilon))$. 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 $f = \mathrm{sign}(\sum_{i=1}^n w_i x_i - \theta)$. 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 [15], hardness of approximation [6, 9], learning Boolean circuits [2], and additive/multiplicative update learning algorithms [17, 7].

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 [20, 16, 3, 18]. 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.* [2] 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 *malicious noise* model. 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 $\eta$ it is an arbitrary pair $(x, y)$ which may be generated by an omniscient adversary. The parameter $\eta$ 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 $\eta$ is greater than $\epsilon/(1+\epsilon)$. Indeed, known algorithms for learning halfspaces [25, 11] or even simpler target functions [19] with malicious noise typically make strong assumptions about the underlying distribution $\mathcal{D}$, and can learn to accuracy $1-\epsilon$ only for noise rates $\eta$ much smaller than $\epsilon$. We describe the most closely related work that we know of in Section 1.2.

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 $\eta = \Omega(\epsilon^2/\log(n/\epsilon))$.*

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 $\eta$ 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* $\eta = \Omega(\epsilon^3/\log(1/\epsilon))$.

### 1.2 Previous Work

Here is some of the most closely related previous work.

**Malicious noise.** General-purpose tools developed by Kearns and Li [12, 13] directly imply that halfspaces can be learned for any distribution over the domain in randomized poly($n$,$1/\epsilon$) time with malicious noise at a rate $\Omega(\epsilon/n)$; the algorithm repeatedly picks a random subsample of the training data, hoping to miss all the noisy examples. Kannan (see [1]) devised a deterministic algorithm with a $\Omega(\epsilon/n)$ bound that repeatedly finds a group of $n+1$ examples that includes a noisy example, then removes the group. Kalai, et al [11] showed that the poly($n$,$1/\epsilon$)-time the averaging algorithm [24] tolerates noise at a rate $\Omega(\epsilon/\sqrt{n})$ when the distribution is uniform. They also described an improvement to $\tilde{\Omega}(\epsilon/n^{1/4})$ based on the observation that uniform examples will tend to be well-separated, so that pairs of examples that are too close to one another can be removed.

**Adversarial label noise.** Kalai, et al showed that if the distribution over the instances is uniform over the unit ball, the averaging algorithm tolerates adversarial label noise at a rate $O(\epsilon/\sqrt{\log(1/\epsilon)})$ in poly($n$,$1/\epsilon$) time. (In that paper, adversarial label noise was called "agnostic learning".) They also described an algorithm that fits low-degree polynomials that tolerates noise at a rate within an additive $\epsilon$ of the accuracy, but in $\mathrm{poly}\left(n^{1/\epsilon^4}\right)$ time; for log-concave distributions, their algorithm took $\mathrm{poly}\left(n^{d(1/\epsilon)}\right)$ time, for an unspecified function $d$. The latter algorithm does not require that the distribution is isotropic, as ours does.

**Robust PCA.** Independently of this work, Xu et al [28] designed and analyzed an algorithm that performs principal component analysis when some of the examples are corrupted arbitrarily, as in the malicious noise model studied here.

### 1.3 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.

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 that 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 scattered 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.

**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 $\epsilon$ 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 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.** 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

[1] We note briefly that the sophisticated outlier removal techniques of [2, 5] 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.

on the ideas of Theorem 1, is sketched in Section 3. In Section 1.2, we described some of the most closely related previous work. Because of space constraints the proof of Theorem 3 is omitted here and is given in the full version [14].

## 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 $\eta$ 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 $A_{\mathrm{mu}}$:

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.* By applying Lagrange multipliers, we can see that the optimal $\mathbf{w}$ is an eigenvector of $A = \sum_{\mathbf{x}\in S} xx^T$. Further, if $\lambda$ is the eigenvalue of $\mathbf{w}$, then $\sum_{\mathbf{x}\in S}(\mathbf{w}\cdot\mathbf{x})^2 = \mathbf{w}^T A\mathbf{w} = \mathbf{w}^T(\lambda\mathbf{w}) = \lambda$. The eigenvector $\mathbf{w}$ with the largest eigenvalue can be found in polynomial time (see e.g. [10]). □

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 $A_{\mathrm{mu}}$. The first says that no direction has much more variance than the expected variance of $1/n$. Its proof, which uses standard tools from VC theory, is omitted due to space constraints.

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

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

The next lemma says that in fact no direction has too many clean examples lying far out in that direction. Its proof, which uses Lemma 7 of [4], is omitted due to space constraints.

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

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

### 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 the any execution of Step 3 in $A_{\mathrm{mu}}$ is at most $6n\log m$.*

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

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

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

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

□

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 $A_{\mathrm{mu}}$ executes step 3, it removes at least $\frac{4m\log m}{n}$ noisy examples from $S_{\mathrm{dirty}}$, 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 $S_{\rm clean}$ of clean examples drawn in step 1, we have

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

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 $A_{\rm mu}$ 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

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

Let us decompose $S_{\rm dirty}$ 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 $|N| \leq |S_{\rm dirty}| \leq \widehat{\eta}m$, (any dirty examples removed in earlier rounds will only reduce the size of $S_{\rm dirty}$) 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 $S = S_{\rm clean}\cup S_{\rm dirty}$ be the sample of $m$ examples drawn from the noisy oracle $\mathrm{EX}_\eta(f,\mathcal{U})$. Let*

- *$S'_{\rm clean}$ be those clean examples that were never removed during step 3 of $A_{\rm mu}$,*
- *$S'_{\rm dirty}$ be those dirty examples that were never removed during step 3 of $A_{\rm mu}$,*
- *$\eta' = \frac{|S'_{\rm dirty}|}{|S'_{\rm clean}\cup S'_{\rm dirty}|}$, i.e. the fraction of dirty examples among the examples that survive step 3, 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.*

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

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

*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 $\eta' \leq o(1)/\log m$ and $\alpha \leq o(1)/\sqrt{n}$, so we shall freely use these bounds in what follows.

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

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

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

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

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

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

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

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

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,

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

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

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

where we have used the triangle inequality and the fact that $\alpha, \eta$ 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 $\alpha$. 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

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

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 $\alpha$ and $\eta'$ 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, with high probability, 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 $\alpha$ 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 $\frac{2n^2}{m/2-n^2}$.

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 $\eta'$ 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 $\epsilon$ 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 $A_{\mathrm{mlc}}$ 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 $\epsilon$, and a bound $\gamma$ 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 during 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. [8]) identifies the properties that we need from a boosting algorithm for our analysis.

[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.

**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 $\gamma$. Then the final output $h$ of $B$ satisfies $\Pr_{x \in \mathcal{D}}[h(x) \neq f(x)] \leq \epsilon$.*

### 3.2 The Algorithm

Our algorithm for learning under isotropic log-concave distributions with malicious noise, Algorithm $A_{\mathrm{mlc}}$, applies the smooth booster from Lemma 7 with the following weak learner, which we call Algorithm $A_{\mathrm{mlcw}}$. (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 $\sum_{(\mathbf{x},y) \in S} (\mathbf{w} \cdot \mathbf{x})^2 \leq c_0 m \log(n/\epsilon)$ then move on to Step 4, 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 $|\mathbf{v} \cdot \mathbf{x}| \leq 3n \log m$, 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. The proof is omitted due to space constraints.

**Lemma 8.** *Suppose Algorithm $A_{\mathrm{mlcw}}$ 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 $A_{\mathrm{mlcw}}$ has advantage $\Omega\left(\frac{\epsilon^2}{n \log(n/\epsilon)}\right)$.*

*Proof Sketch.* We exploit the fact that isotropic logconcave distributions are not *very* concentrated to show that clean examples tend to be classified correctly by a large margin. We then use concentration bounds to prove analogs of Lemmas 2 and 3, and put them together in a roughly similar way. □

## References


[1] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 724–733, 1993.

[2] 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.

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

[4] 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.

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

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

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

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

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

[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] M. Kearns, R. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2/3):115–141, 1994.

[14] A. Klivans, P. Long, and R. Servedio. Learning Halfspaces with Malicious Noise. Full version available at http://www.cs.columbia.edu/˜rocco/papers/icalp09malicious.html, 2009.

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

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

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

[18] 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.

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

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

[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.

[28] H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. *JMLR*, 2009. to appear.