Agnostically Learning Halfspaces

Adam Tauman Kalai
TTI-Chicago
kalai@tti-c.org

Adam R. Klivans*
UT-Austin
klivans@cs.utexas.edu

Yishay Mansour*
Tel Aviv University
mansour@cs.tau.ac.il

Rocco A. Servedio*†
Columbia University
rocco@cs.columbia.edu

January 26, 2006

Abstract

We give the first dimension-efficient algorithm that learns (under distributional assumptions) a halfspace in the difficult *agnostic* framework of Kearns *et al.* [21], where a learner is given access to a distribution on labelled examples but the labelling may be arbitrary (similar to malicious noise). It constructs a hypothesis whose error rate on future examples is within an additive ϵ of the optimal halfspace, in time poly(n) for any constant $\epsilon > 0$, for the uniform distribution over $\{-1, 1\}^n$ or unit sphere in $\mathbb{R}^n$, as well as any log-concave distribution in $\mathbb{R}^n$. It also agnostically learns Boolean disjunctions in time $2^{\tilde{O}(\sqrt{n})}$ with respect to *any* distribution. The L_1 polynomial regression algorithm is a natural noise-tolerant arbitrary-distribution generalization of the well known "low-degree" Fourier algorithm of Linial, Mansour, & Nisan. We observe that significant improvements on the running time of our algorithm would yield the fastest known algorithm for learning parity with noise, a challenging open problem in computational learning theory.

Additionally, we obtain a new algorithm for PAC learning halfspaces under the uniform distribution on the unit sphere which tolerates more *malicious noise* than previous algorithms.

1 Introduction

Halfspaces have been used extensively in Machine Learning for decades. From the early work on the Perceptron algorithm in the 50's, through the learning of artificial neural networks in the 80's, and up to and including today's Adaboost [12] and Support Vector Machines [38], halfspaces have played a central role in the development of the field's most important tools.

Formally, a *halfspace* is a Boolean function $f(x) = \text{sgn}(\sum_{i=1}^{n} w_i x_i - \theta)$. While efficient algorithms are known for learning halfspaces if the data is guaranteed to be noise-free, learning a halfspace from noisy examples remains a challenging and important problem. Halfspace-based learning methods appear repeatedly in both theory and practice, and they are frequently applied to labeled data sets which are not linearly separable. This motivates the following natural and well-studied question: what can one *provably* say about the performance of halfspace-based learning methods in the presence of noisy data or distributions that do not obey constraints induced by an unknown halfspace? Can we develop learning algorithms which tolerate data generated from a "noisy" halfspace and output a meaningful hypothesis?

---

*Some of this research done while visiting TTI-Chicago.

†Supported in part by NSF CAREER award CCF-0347282 and a Sloan Foundation fellowship.

1.1 Agnostic Learning.

The *agnostic learning* framework, introduced by Kearns et al. [21], is an elegant model for studying the phenomenon of learning from noisy data. In this model the learner receives labeled examples (x, y) drawn from a fixed distribution over example-label pairs, but (in contrast with Valiant's standard PAC learning model [36]) the learner cannot assume that the labels y are generated by applying some target function f to the examples x. Of course, without any assumptions on the distribution it is impossible for the learner to always output a meaningful hypothesis. Kearns *et al.* instead require the learner to output a hypothesis whose accuracy with respect to future examples drawn from the distribution approximates that of the optimal concept from some fixed concept class of functions $\mathcal{C}$, such as the class of all halfspaces $f(x) = \mathrm{sgn}(v \cdot x - \theta)$. Given a concept class $\mathcal{C}$ and a distribution $\mathcal{D}$ over labeled examples (x, y), we write $\mathsf{opt} = \min_{f \in \mathcal{C}} \Pr_{\mathcal{D}}[f(x) \neq y]$ to denote the error rate of the optimal (smallest error) concept from $\mathcal{C}$ with respect to $\mathcal{D}$.

For intuition, one can view agnostic learning as a noisy learning problem in the following way: There is a distribution $\mathcal{D}$ over examples x and the data *is* assumed to be labeled according to a function $f \in \mathcal{C}$, but an adversary is allowed to corrupt an $\eta = \mathsf{opt}$ fraction of the labels given to the learning algorithm. The goal is find a hypothesis h with error $\Pr_{\mathcal{D}}[h(x) \neq y]$ as close as possible to η. (We note that such a noise scenario is far more challenging than the *random classification noise* model, in which an η fraction of labels are flipped independently at random and for which a range of effective noise-tolerant learning algorithms are known [19, 4].)

Unfortunately, only few positive results are known for agnostically learning expressive concept classes. Kearns *et al.* [21] gave an algorithm for agnostically learning piecewise linear functions, and Goldman *et al.* [14] showed how to agnostically learn certain classes of geometric patterns. Lee *et al.* [24] showed how to agnostically learn some very restricted classes of neural networks in time exponential in the fan-in. On the other hand, some strong negative results are known: in the case of *proper learning* (where the output hypothesis must belong to $\mathcal{C}$), agnostic learning is known to be NP-hard even for the concept class $\mathcal{C}$ of disjunctions [21]. In fact, it is known [25] that agnostically learning disjunctions, even with *no* restrictions on the hypotheses used, is at least as hard as PAC learning DNF formulas, a longstanding open question in learning theory.

Thus, it is natural to consider, as we do in this paper, agnostic learning with respect to various restricted distributions $\mathcal{D}$ for which the marginal distribution $\mathcal{D}_X$ over the example space X satisfies some prescribed property. This corresponds to a learning scenario in which the *labels* are arbitrary but the distribution over *examples* is restricted.

1.2 Our Main Technique.

The following two observations are the starting point of our work:

- The "low-degree" Fourier learning algorithm of Linial *et al.* can be viewed as an algorithm for performing L_2-norm *polynomial regression* under the uniform distribution on $\{-1, 1\}^n$. (See Section 2.2.)

- A simple analysis (Observation 3) shows that the low-degree algorithm has some attractive agnostic learning properties under the uniform distribution on $\{-1, 1\}^n$. (See Section 2.3.)

The "low-degree" algorithm, however, will only achieve partial results for agnostic learning (the output hypothesis will be within a factor of 8 of optimal). As described in Section 3, the above two observations naturally motivate a new algorithm which can be viewed as an L_1-norm version of the low-degree algorithm; we call this simply the *polynomial regression algorithm.* (At this point it may be slightly mysterious why the L_1 norm would be significantly better than the L_2 norm; we discuss this point in Section 3.)

Roughly speaking our main result about the polynomial regression algorithm, Theorem 5, shows the following (see Section 3 for the detailed statement):

> Given a concept class $\mathcal{C}$ and a distribution $\mathcal{D}$, if concepts in $\mathcal{C}$ can be approximated by low-degree polynomials in the L_2-norm relative to the marginal distribution $\mathcal{D}_X$, then the L_1 polynomial regression algorithm is an efficient agnostic learning algorithm for $\mathcal{C}$ with respect to $\mathcal{D}$.

A long line of research has focused on how well the truncated Fourier polynomial over the parity basis approximates concept classes with respect to the L_2 norm; this has led to numerous algorithms for learning concepts with respect to the uniform distribution over the Boolean hypercube $\{-1,1\}^n$ [26, 8, 16, 18, 22]. For learning with respect to the uniform distribution on the unit sphere, our analysis uses the Hermite polynomials [35], a family of orthogonal polynomials with a weighting scheme related to the density function of the Gaussian distribution. As such, these polynomials are well suited for approximating concepts with respect to the L_2 norm over S^{n-1}. We believe this approach will find further applications in the future.

Additionally, we show that a slightly modified version of the wildly popular *Support Vector Machine* (SVM) algorithm [38], with a polynomial kernel, can achieve the same result[1]. Unfortunately, with the number of examples we require for our analysis, the SVM algorithm is no more efficient than our simple polynomial regression algorithm (the "Kernel trick" does not help). But it is interesting to give strong provable guarantees about the agnostic learning ability of an algorithm that is so popular in practice.

1.3 Our Main Results.

As described below, our main result about the polynomial regression algorithm can be applied to obtain many results for agnostic learning of halfspaces with respect to a number of different distributions, both discrete and continuous, some uniform and some nonuniform.

Theorem 1 *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^n \times \{-1,1\}$. The L_1 polynomial regression algorithm has the following properties: its runtime is polynomial in the number of examples it is given, and*

1. *If the marginal $\mathcal{D}_X$ is (a) uniform on $\{-1,1\}^n$* or *(b)* uniform on the unit sphere in *$\mathbb{R}^n$, then with probability $1-\delta$ the polynomial regression algorithm outputs a hypothesis with error $\mathsf{opt}+\epsilon$ given $\mathrm{poly}(n^{1/\epsilon^4}, \log\frac{1}{\delta})$ examples.*

2. *If the marginal $\mathcal{D}_X$ is* log-concave*, then with probability $1-\delta$ the polynomial regression algorithm outputs a hypothesis with error $\mathsf{opt}+\epsilon$ given $\mathrm{poly}(n^{d(\epsilon)}, \log\frac{1}{\delta})$ examples, where $d : \mathbb{R}_+ \to \mathbb{Z}_+$ is a universal function independent of $\mathcal{D}_X$ or n.*

Part 1(a) follows from our analysis of the L_1 polynomial regression algorithm combined with the Fourier bounds on halfspaces given by Klivans *et al.* [22]. Part 1(b) follows from the same analysis of the algorithm combined with concentration bounds over the n-dimensional sphere. In proving such bounds, we use the Hermite polynomial basis in analogy with the Fourier basis used

[1]This was pointed out to us by Avrim Blum.

previously. (We note that learning halfspaces under the uniform distribution on S^{n-1} is a well-studied problem, see e.g. [1, 2, 19, 27, 28].) As before, we show that a related algorithm gives a hypothesis with error $O(\mathsf{opt} + \epsilon)$ in time $n^{O(1/\epsilon^2)}$.

In Section 4.2, we show that algorithms for agnostically learning halfspaces with respect to the uniform distribution on $\{0,1\}^n$ can be used to solve the well-known problem of learning parity functions with respect to random classification noise [5]. This indicates that substantially improving the results of part (1) may be very difficult. For example, even an $n^{O(1/\epsilon^{2-\beta})}$ time algorithm ($\beta > 0$) for agnostically learning halfspaces (with respect to the uniform distribution over the hypercube) would yield the fastest known algorithm for learning parity with noise.

As indicated by part (2) of Theorem 2, for any constant ϵ, We can also achieve a polynomial-time algorithm for learning with respect to any log-concave distribution. Recall that any Gaussian distribution, exponential distribution, and uniform distribution over a convex set is log-concave.

We next consider a simpler class of halfspaces: disjunctions on n variables. The problem of agnostically learning an unknown disjunction (or learning noisy disjunctions) has long been a difficult problem in computational learning theory and was recently re-posed as a challenge by Avrim Blum in his FOCS 2003 tutorial [3]. By combining Theorem 5 with known constructions of low-degree polynomials that are good L_∞-approximators of the OR function, we obtain a subexponential time algorithm for agnostically learning disjunctions with respect to *any* distribution (recall that since this problem is at least as hard as PAC-learning DNF, given the current state of the art we do not expect to achieve a polynomial-time algorithm):

Theorem 2 *Let $\mathcal{D}$ be a distribution on $X \times Y$ where $\mathcal{D}$ is an arbitrary distribution over $\{-1,1\}^n$ and $Y = \{-1,1\}$. For the class of disjunctions, with probability $1-\delta$ the polynomial regression algorithm outputs a hypothesis with error $\leq \mathsf{opt} + \epsilon$ in time $2^{\tilde{O}(\sqrt{n}\cdot\log(1/\epsilon))} \cdot \mathrm{poly}(\log \frac{1}{\delta})$.*

1.4 Extensions and Other Applications.

We believe that the polynomial regression algorithm will have many extensions and applications; so far we have only explored a few of these which we now describe.

In Section 4.3 we show how our approach can be used to improve the algorithm due to Klivans *et al.* [22] for learning intersections of halfspaces with respect to the uniform distribution over the hypercube.

In Section 5 we give a detailed analysis of an algorithm which is essentially the same as the degree-1 version of the polynomial regression algorithm, for agnostic learning the concept class of origin-centered halfspaces $\mathrm{sgn}(v \cdot x)$ over the uniform distribution on the sphere S^{n-1}. While our analysis from Section 3 only implies that this algorithm should achieve some fixed constant error $\Theta(1)$ independent of opt, we are able to show that in fact we do much better if opt is small:

Theorem 3 *Let $\mathcal{D}$ be a distribution on $X \times Y$, where $Y = \{-1,1\}$ and the marginal $\mathcal{D}_X$ is uniform on the sphere S^{n-1} in $\mathbb{R}^n$. There is a simple algorithm for agnostically learning origin-centered halfspaces with respect to $\mathcal{D}$ which uses $m = O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$ examples, runs in $\mathrm{poly}(n, 1/\epsilon, \log\frac{1}{\delta})$ time, and outputs a hypothesis with error $O(\mathsf{opt}\sqrt{\log\frac{1}{\mathsf{opt}}} + \epsilon)$.*

This result thus trades off accuracy versus runtime compared with Theorem 1. We feel that Theorem 3 is intriguing since it suggests that a deeper analysis might yield improved runtime bounds for Theorem 1 as well.

In Section 6 we consider the problem of learning an unknown origin-centered halfspace under the uniform distribution on S^{n-1} in the presence of *malicious noise* (we give a precise definition of the malicious noise model in Section 6). Recall from Section 1.1 that we can view agnostic learning with respect to a particular marginal distribution $\mathcal{D}_X$ as the problem of learning under $\mathcal{D}_X$ in the presence of an adversary who may change the *labels* of an η fraction of the examples, without changing the actual distribution $\mathcal{D}_X$ over examples. In contrast, in the model of learning under *malicious noise* with respect to $\mathcal{D}_X$, roughly speaking the adversary is allowed to change an η fraction of the labels *and examples* given to the learner. As described in Section 6 this is a very challenging noise model in which only limited positive results are known. We show that by combining the algorithm of Theorem 3 with a simple preprocessing step, we can achieve relatively high tolerance to malicious noise:

Theorem 4 *There is a simple algorithm for learning origin-centered halfspaces under the uniform distribution on S^{n-1} to error ϵ in the presence of malicious noise when the noise rate η is at most $O(\frac{\epsilon}{n^{1/4}\log^{1/4}(n/\epsilon)})$. The algorithm runs in $\mathrm{poly}(n, 1/\epsilon, \log\frac{1}{\delta})$ time and uses $m = O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$ many examples.*

This is the highest known rate of malicious noise that can be tolerated in polynomial time for any nontrivial halfspace learning problem. The preprocessing step can be viewed as a somewhat counterintuitive form of outlier removal – instead of identifying and discarding examples that lie "too far" from the rest of the data set, we discard examples that lie too *close* to any other data point. The analysis of this approach relies on classical results from sphere packing.

2 Preliminaries

Let $\mathcal{D}$ be an arbitrary distribution on $X \times \{-1, 1\}$, for some set X. Let $\mathcal{C}$ be a class of Boolean functions on X. Define the *error of* $f : X \to \{-1, 1\}$ and the *optimal error of* $\mathcal{C}$ to be

$$\mathsf{err}(f) = \Pr_{(x,y)\leftarrow\mathcal{D}}[f(x) \neq y], \quad \mathsf{opt} = \min_{c\in\mathcal{C}} \mathsf{err}(c),$$

respectively. Roughly speaking, the goal in agnostic learning of a concept class $\mathcal{C}$ is as follows: given access to examples drawn from distribution $\mathcal{D}$, we wish to efficiently find a hypothesis with error not much larger than opt. More precisely, we say $\mathcal{C}$ is *agnostically learnable* if there exists an algorithm which takes as input δ, ϵ, and has access to an example oracle $\mathsf{EX}(\mathcal{D})$ and outputs with probability greater than $1 - \delta$ a *hypothesis* $h : X \to \{-1, 1\}$ such that $\mathsf{err}(h) \leq \mathsf{opt} + \epsilon$. We say $\mathcal{C}$ is agnostically learnable in time t if its running time (including calls to the example oracle) is bounded by $t(\epsilon, \delta, n)$. If the above only holds for a distribution $\mathcal{D}$ whose margin is uniform over X, we say the algorithm *agnostically learns* $\mathcal{C}$ *over the uniform distribution.* See [21] for a detailed description of the agnostic learning framework.

A distribution is log-concave if its support is convex and it has a probability density function whose logarithm is a concave function from $\mathbb{R}^n$ to $\mathbb{R}$.

In all our algorithms we assume that we are given m labeled examples $\mathcal{Z} = (x^1, y^1), \ldots, (x^m, y^m)$ drawn independently from the distribution $\mathcal{D}$ over $X \times \{-1, 1\}$. The $\mathrm{sgn} : \mathbb{R} \to \{-1, 1\}$ function is defined by $\mathrm{sgn}(z) = 1$ if $z \geq 0$, $\mathrm{sgn}(z) = -1$ if $z < 0$.

2.1 Fourier preliminaries and the low-degree algorithm.

For $S \subseteq [n]$ the parity function $\chi_S : \{-1, 1\}^n \to \{-1, 1\}$ over the variables in S is simply the multilinear monomial $\chi_S(x) = \prod_{i\in S} x_i$. The set of all 2^n parity functions $\{\chi_S\}_{S\subseteq[n]}$ forms an orthonormal basis for the vector space of real-valued functions on $\{-1, 1\}^n$, with respect to the inner product $(f, g) = \mathbf{E}[fg]$ (here and throughout Section 2.1 unless otherwise indicated all probabilities and expectations are with respect to the uniform distribution over $\{-1,1\}^n$). Hence every real-valued function $f : \{-1,1\}^n \to \mathbb{R}$ can be uniquely expressed as a linear combination

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x). \tag{1}$$

The coefficients $\hat{f}(S) = \mathbf{E}[f\chi_S]$ of the Fourier polynomial (1) are called the *Fourier coefficients* of f; collectively they constitute the *Fourier spectrum* of f. We recall *Parseval's identity*, which states that for every real-valued function f we have $\mathbf{E}[f(x)^2] = \sum_S \hat{f}(S)^2$. For Boolean functions we thus have $\sum_S \hat{f}(S)^2 = 1$.

The "low-degree algorithm" for learning Boolean functions under the uniform distribution via their Fourier spectra was introduced by Linial *et al.* [26], and has proved to be a powerful tool in uniform distribution learning. The algorithm works by empirically estimating each coefficient $\hat{f}(S) \approx \tilde{f}(S) := \frac{1}{m}\sum_{j=1}^m f(x^j)\chi_S(x^j)$ with $|S| \le d$ from the data, and constructing the degree-d polynomial $p(x) = \sum_{|S|\le d} \tilde{f}(S)\chi_S(x)$ as an approximation to f. (Note that the polynomial $p(x)$ is real-valued rather than Boolean-valued. If a Boolean-valued classifier h is desired, it can be obtained by taking $h(x) = \mathrm{sgn}(p(x))$, and using the simple fact $\Pr_{\mathcal{D}}[\mathrm{sgn}(p(x)) \ne g(x)] \le \mathbf{E}_{\mathcal{D}}[(p(x) - f(x))^2]$ which holds for any polynomial p, any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, and any distribution $\mathcal{D}$.)

Let $\alpha(\epsilon, n)$ be a function $\alpha : (0, 1/2) \times \mathsf{N} \to \mathsf{N}$. We say that concept class $\mathcal{C}$ has a *Fourier concentration bound* of $\alpha(\epsilon, n)$ if, for all $n \ge 1$, all $0 < \epsilon < \frac{1}{2}$, and all $f \in \mathcal{C}_n$ we have $\sum_{|S|\ge\alpha(\epsilon,n)} \hat{f}(S)^2 \le \epsilon$. The low-degree algorithm is useful because it efficiently constructs a high-accuracy approximator for functions that have good Fourier concentration bounds (we suppress the logarithmic dependence on the failure probability δ to improve readability):

Fact 1 ([26]) *Let $\mathcal{C}$ be a concept-class with concentration bound $\alpha(\epsilon/2, n)$. Then for any $f \in \mathcal{C}$, given data labeled according to f and drawn from the uniform distribution on $X = \{-1,1\}^n$, the low-degree algorithm outputs, with probability $1 - \delta$, a polynomial p such that $\mathbf{E}[(p(x) - f(x))^2] \le \epsilon$ and runs in time $\mathrm{poly}(n^{\alpha(n,\epsilon)}, \log \frac{1}{\delta})$.*

The idea behind Fact 1 is simple: if the coefficients of p were precisely $\hat{f}(S)$ instead of $\tilde{f}(S)$, then the Fourier concentration bound and Parseval's identity would give $\sum_{|S|\ge\alpha(\epsilon/2,n)} = \mathbf{E}[(p(x) - f(x))^2] \le \epsilon/2$. The extra $\epsilon/2$ is incurred because of approximation error in the estimates $\tilde{f}(S)$.

2.2 The low-degree algorithm and L_2 polynomial regression.

The main observation of this section is that the low-degree Fourier algorithm of [26] can be viewed as a special case of least-squares polynomial regression over uniform distributions on the n-dimensional cube.

Let $\mathcal{D}$ be a distribution over $X \times \{-1,1\}$. In least-squares (L_2-norm) polynomial regression, one attempt to minimize the following:

$$\min_{\deg(p)\le d} \mathbf{E}_{\mathcal{D}}\left[(p(x) - y)^2\right] \approx \min_{\deg(p)\le d} \frac{1}{m}\sum_{j=1}^m \left(p(x^j) - y^j\right)^2. \tag{2}$$

Ideally, one would like to minimize the LHS, i.e. find the best degree d polynomial L_2 approximation to y over $\mathcal{D}$. To do this (approximately) given a data set, we minimize the right-hand side. In particular, we write a polynomial as a sum over all degree $\le d$ monomials, $p(x) = \sum_b p_b \prod_{i=1}^n (x_i)^{b_i}$

where the sum is over $\{b \in \mathbb{Z}^n \mid \sum_{i=1}^n b_i \le d, \forall i\ b_i \ge 0\}$. In turn, this can be viewed as a standard *linear* regression problem if we expand example x^j into a vector with a coordinate $\prod_{i=1}^n (x_i^j)^{b_i}$, for each of the $\le n^{d+1}$ different b's. Least-squares linear regression, in turn, can be solved by a single matrix inversion; and thus in general we can approximate the RHS of (2) in $n^{O(d)}$ time.

Now let us consider L_2 polynomial regression in the uniform distribution scenario where $X = \{-1,1\}^n$, $y = f(x)$ for some function $f : X \to \{-1,1\}$, and we have a uniform distribution $\mathcal{U}_X$ over $x \in \{-1,1\}^n$. Since $x^2 = 1$ for $x \in \{-1,1\}$, we may consider only degree-d multilinear polynomials, i.e. sums of monomials $\chi_S(x) = \prod_{i \in S} x_i$ with $S \subseteq [n], |S| \le d$. Using Parseval's identity, it is not difficult to show that best degree d polynomial is exactly

$$\arg\min_{\deg(p) \le d} \mathbf{E}_{\mathcal{U}_X}\left[(p(x) - f(x))^2\right] = \sum_{S \subseteq [n]:|S| \le d} \hat{f}(S)\chi_S(x),$$

where $\hat{f}(S) = \mathbf{E}_{\mathcal{U}_X}[f(x)\chi_S(x)]$. Thus in this uniform case, one can simply estimate each coefficient $\hat{f}(S) \approx \frac{1}{m}\sum_{j=1}^m f(x^j)\chi_S(x^j)$ rather than solving the general least-squares regression problem; and this is precisely what the low-degree algorithm does.

In the *nonuniform* case, it is natural to consider running general L_2 polynomial regression rather than the low-degree algorithm. We do something similar to this in Section 3, but first we consider the *agnostic* learning properties of the low-degree algorithm in the next subsection.

2.3 Using the low-degree algorithm as an agnostic learner.

Kearns *et al.* prove the following statement about agnostic learning with the low-degree algorithm:

Fact 2 ([21], Corollary 1) *Let $\mathcal{C}$ be a concept class with concentration bound $\alpha(\epsilon, n)$. Then the low-degree algorithm agnostically learns $\mathcal{C}$ under the uniform distribution to error* $\frac{1}{2} - (\frac{1}{2} - \mathsf{opt})^2 + \epsilon = \frac{1}{4} + \mathsf{opt}(1 - \mathsf{opt}) + \epsilon$ *with probability* $1 - \delta$ *and in time* $\mathrm{poly}(n^{\alpha(\epsilon/2,n)}, \log\frac{1}{\delta})$.

This was termed a "weak agnostic learner" in [21] because as long as opt is bounded away from 1/2, this resulting hypothesis has error bounded from 1/2. We now show that the low-degree algorithm is in fact a "strong agnostic learner," in that if opt is small it can in fact achieve very low error:

Observation 3 *Let $\mathcal{C}$ be a concept class with concentration bound $\alpha(\epsilon, n)$. Then the low-degree algorithm agnostically learns $\mathcal{C}$ under the uniform distribution to error* $8\mathsf{opt} + \epsilon$ *in time* $n^{O(\alpha(\epsilon/2,n))}$.

Proof: Let $f \in C$ be an optimal function, i.e. $\Pr[y \ne f(x)] = \mathsf{opt}$. As described above, the low-degree algorithm (approximately) finds the best degree-d approximation $p(x)$ to the data y, i.e. $\min_{\deg(p) \le d} \mathbf{E}[(p(x) - y)^2]$, and the same term represents the mean squared error of p. This can be bounded using the "almost-triangle" inequality $(a - c)^2 \le 2\left((a-b)^2 + (b-c)^2\right)$ for $a, b, c \in \mathbb{R}$.

$$\begin{aligned}
\min_{\deg p \le d} \mathbf{E}[(y - p(x))^2] &\le \mathbf{E}\left[\left(y - \textstyle\sum_{|S|<d} \hat{f}(S)\chi_S(x)\right)^2\right] \\
&\le 2\mathbf{E}\left[(y - f(x))^2 + \left(f(x) - \textstyle\sum_{|S|<d} \hat{f}(S)\chi_S(x)\right)^2\right] \\
&= 2\left(4\Pr[y \ne f(x)] + \textstyle\sum_{|S| \ge d} \hat{f}(S)^2\right)
\end{aligned}$$

The first term is $8\mathsf{opt}$ and the second is at most $\epsilon/2$ for $d = \alpha(n, \epsilon/2)$, where an additional $\epsilon/2$ is due to the sampling. Outputting $h(x) = \mathrm{sgn}(p(x))$ gives error at most $8\mathsf{opt} + \epsilon$ because $\Pr[\mathrm{sgn}(p(x)) \ne y] \le \mathbf{E}[(p(x) - y)^2]$. ■

Another way to state this is that if f and $\tilde{f}$ are two functions and f has a Fourier concentration bound of $\alpha(\epsilon, n)$, then $\tilde{f}$ satisfies the concentration bound $\sum_{|S| \ge \alpha(n,\epsilon)} \hat{\tilde{f}}(S)^2 \le 8\Pr[f(x) \ne \tilde{f}(x)] + 2\epsilon$.

The L_1 polynomial regression algorithm$(\mathcal{Z} = (x^1, y^1) \dots (x^m, y^m), d)$**:**

1. Find polynomial p of degree $\leq d$ to minimize $\frac{1}{m}\sum_{j=1}^{m}|p(x^j) - y^j|$. (This can be done by expanding examples to include all monomials of degree $\leq d$ and then performing L_1 *linear* regression, as described earlier.)
2. Output $h(x) = \mathrm{sgn}\,(p(x) - t)$ where $t \in [-1, 1]$ is chosen so as to minimize the error of the hypothesis on $\mathcal{Z}$.

3 L$_1$ polynomial regression

Given the setup in the previous sections, it is natural to expect that we will now show that the general L_2 polynomial regression algorithm has agnostic learning properties similar to those established for the low-degree algorithm in Observation 3. However, such an approach only yields error bounds of the form $O(\mathsf{opt}+\epsilon)$, and for agnostic learning our real goal is a bound of the form $\mathsf{opt}+\epsilon$. To achieve this, we will instead use L_1-*norm*, rather than L_2-norm.

Analogous to (2), in L_1-norm polynomial regression we attempt to minimize:

$$\min_{\deg(p)\leq d} \mathbf{E}_{\mathcal{D}}\left[|p(x) - y|\right] \approx \min_{\deg(p)\leq d} \frac{1}{m}\sum_{j=1}^{m}\left|p(x^j) - y^j\right|. \tag{3}$$

To solve the RHS minimization problem, again each example is expanded into a vector of length $\leq n^{d+1}$ and an algorithm for L_1 linear regression is applied. L_1 linear regression is a well-studied problem, and the minimizing polynomial p for the RHS of (3) can be obtained in $\mathrm{poly}(n^d)$ time using linear programming (see Appendix A for an elaboration on this point). For our purposes we will be satisfied with an approximate minimum, and hence one can use a variety of techniques for approximately solving linear programs efficiently.

How do L_1 and L_2 polynomial regression compare? In the noiseless case, both (2) and (3) approach 0 at related rates as d increases. However, in the noisy/agnostic case, flipping the sign of $y = \pm 1$ changes $(p(x) - y)^2$ by $4p(x)$ which can potentially be very large; in contrast, flipping y's sign can only change $|p(x) - y|$ by 2. On the other hand, it is often easier to bound the L_1 error in terms of the mathematically convenient L_2 error. Thus while our polynomial regression algorithm works only with the L_1 norm, the performance bound and analysis depends on the L_2 norm.

3.1 The algorithm and proof of correctness.

We now give the polynomial regression algorithm and establish conditions under which it is an agnostic learner achieving error $\mathsf{opt}+\epsilon$. The algorithm takes as input m examples, $\mathcal{Z} = (x^1, y^1) \dots (x^m, y^m)$ and a degree d.

Theorem 5 *Suppose* $\min_{\deg(p)\leq d} E_{\mathcal{D}_X}[(p(x) - c(x))^2] \leq \epsilon^2$ *for some degree d, some distribution $\mathcal{D}$ over $X \times \{-1, 1\}$ with marginal $\mathcal{D}_X$, and any c in the concept class $\mathcal{C}$. Then, for h output by the degree-d L_1 polynomial regression algorithm with $m = \mathrm{poly}(n^d/\epsilon)$ examples, $\mathbf{E}_{\mathcal{Z}}[\mathsf{err}(h)] \leq \mathsf{opt} + \epsilon$.*

If we repeat the same algorithm $r = O(\log(1/\delta)/\epsilon)$ times with fresh examples each, and let h be the hypothesis with lowest error on an independent test set of size $O(\log(1/\delta)/\epsilon^2)$, then with probability $1 - \delta$, $\mathsf{err}(h) \leq \mathsf{opt} + \epsilon$.

Remark 4 *Note that using Theorem 5, a Fourier concentration bound of $\alpha(n, \epsilon)$ immediately implies that the L_1 regression algorithm achieves error $\mathsf{opt} + \epsilon$ in time $n^{O(\alpha(n,\epsilon^2))}$ for distributions $\mathcal{D}$*

with marginal $\mathcal{D}_X$ that is uniform on $\{-1,1\}^n$. As we will see in the next section, Theorem 5 can be applied to other distributions as well.

Proof of Theorem 5: Suppose the algorithm chooses polynomial p and threshold t. First, we claim that the empirical error of h on $\mathcal{Z}$ is at most one half the L_1 error of p:

$$\frac{1}{m}\sum_{j=1}^{m} \mathrm{I}(h(x^j) \neq y^j) \leq \frac{1}{2m}\sum_{j=1}^{m} \left|y^j - p(x^j)\right|. \tag{4}$$

To see this, note that $h(x^j) \neq y^j$ if and only if the threshold $t \in [-1,1]$ lies in between the numbers $p(x^j)$ and y^j, i.e., if they are on the same side of t then $\mathrm{sgn}(p(x^j)-t) = \mathrm{sgn}(y^j - t) = y^j$. Hence, even if we chose a *uniformly random* $t \in [-1,1]$, for any j, the chance of t splitting these numbers is at most $\left|y^j - p(x^j)\right|/2$ because the width of $[-1,1]$ is 2 and the separation between the numbers is $\left|y^j - p(x^j)\right|$. Thus, (4) holds in expectation for random $t \in [-1,1]$. Since the algorithm chooses t to maximize the LHS of (4), it holds with certainty. (This reduction is a general procedure for converting an L_1 bound on error to a classification error and a similar randomized threshold idea was used by Blum *et al.* for the low-degree algorithm [6].)

Let c be an optimal classifier in $\mathcal{C}$, and let p^* be a polynomial with $\mathbf{E}_{\mathcal{D}}[(c(x)-p^*(x))^2] \leq \epsilon^2$. By the fact that $E[|Z|] \leq \sqrt{E[Z^2]}$ for any random variable Z, we have $\mathbf{E}_{\mathcal{D}}[|c(x)-p^*(x)|] \leq \epsilon$. By the algorithm's choice of p, we have,

$$\frac{1}{m}\sum_{j=1}^{m} \left|y^j - p(x^j)\right| \leq \frac{1}{m}\sum_{j=1}^{m} \left|y^j - p^*(x^j)\right| \leq \frac{1}{m}\sum_{j=1}^{m} \left|y^j - c(x^j)\right| + \left|c(x^j) - p^*(x^j)\right|.$$

The expectation of the RHS is $\leq 2\mathsf{opt} + \epsilon$. Taking expectations and combining with (4) gives,

$$\mathbf{E}_{\mathcal{Z}}\left[\frac{1}{m}\sum_{j=1}^{m} \mathrm{I}(h(x^j) \neq y^j)\right] \leq \mathsf{opt} + \frac{\epsilon}{2}.$$

By VC theory, for $m = \mathrm{poly}(n^d/\epsilon)$ examples, the empirical error $\frac{1}{m}\sum_{j=1}^{m} \mathrm{I}(h(x^j) \neq y^j)$ above and generalization error $\mathsf{err}(h)$ will be differ by at most an expected $\epsilon/4$. Hence, the first part of the Theorem is implied by,

$$\mathbf{E}_{\mathcal{Z}}\left[\mathsf{err}(h)\right] \leq \mathsf{opt} + (3/4)\epsilon.$$

The second part of the theorem is a relatively standard reduction from expected error to high-probability guarantees. In particular, by Markov's inequality, on any single repetition,

$$\Pr_{\mathcal{Z}}\left[\mathsf{err}(h) \geq \mathsf{opt} + (7/8)\epsilon\right] \leq \frac{\mathsf{opt} + (3/4)\epsilon}{\mathsf{opt} + (7/8)\epsilon} \leq 1 - \frac{\epsilon}{16}.$$

Hence, after $r = O(\log(1/\delta)/\epsilon)$ repetitions of the algorithm, with probability $1-\delta/2$, one of them will have $\mathsf{err}(h) \leq \mathsf{opt} + (7/8)\epsilon$. In this case, using an independent set of size $O(\log(1/\delta)/\epsilon^2)$, with probability at most $\delta/2$, we will choose one with error $> \mathsf{opt} + \epsilon$. ■

As noted at the very beginning of this section, an analogous L_2 algorithm could be defined to minimize $\frac{1}{m}\sum_{j=1}^{m}(p(x^j) - y^j)^2$ rather than $\frac{1}{m}\sum_{j=1}^{m}|p(x^j) - y^j|$. Error guarantees of the form $O(\mathsf{opt} + \epsilon)$ can be shown for this L_2 algorithm, following the same argument but again using the "almost-triangle" inequality.

3.2 Relationship to SVMs

As pointed out by Avrim Blum, our algorithm is very similar to an SVM with a polynomial kernel and can be made even more similar. The standard Support Vector Machine with a degree-d *polynomial kernel* solves the following minimization problem:

$$\min_{\deg(p)\le d}(1-\lambda)\frac{1}{m}\sum_{i=1}^{m}L(y^i,z)+\lambda(\text{regularization term}),$$

where $L(y^i,z)=\max\{0,1-y^iz\}$. It does this using an algorithmic trick that requires time only $\text{poly}(m,n,d)$. In theory, this could be substantially faster than our $n^{O(d)}$ algorithm. However, for our analysis, we require $m=n^{O(d)}$ samples, in which case the SVM algorithm is no faster.

Step 1 of our algorithm could be replaced by the above minimization problem, with $\lambda=0$, and the analysis would hold almost exactly as is. Intuitively, this is because, for $|y|=1$, $L(y,z)=|y-z|$ unless $yz>1$. However, if $yz>1$, thresholding z with $t\in[-1,1]$ will certainly give us the correct prediction of this y. More technically, we have that, for $|y|=1$, $L(y,z)\le|y-z|$, yet we still have that $\Pr_{t\in[-1,1]}[y\ne\text{sgn}(z-t)]\le\frac{1}{2}L(y,z)$ (we now have $L(y,z)$ where we had $|y-z|$).

Hence one can use a standard SVM package to implement our algorithm, setting the regularization parameter to 0. The only nonstandard part would be choosing an optimal threshold t rather than using standard SVM choice of $t=0$.

4 Agnostic learning halfspaces and disjunctions via polynomial regression

In this section we show how to apply Theorem 5 to prove Theorems 1 and 2.

As noted in Remark 4, Theorem 5 implies that any concept class with a Fourier concentration bound is in fact agnostically learnable to error $\mathsf{opt}+\epsilon$ under the uniform distribution on $\{-1,1\}^n$. In particular, Theorem 1 1(a) follows immediately from the Fourier concentration bound for halfspaces of [22]:

Fact 5 *[22] The concept class $\mathcal{C}$ of all halfspaces over $\{-1,1\}^n$ has a Fourier concentration bound of $\alpha(\epsilon,n)=441/\epsilon^2$.*

For the uniform distribution on S^{n-1} and any log-concave distribution, we can prove the existence of a good low-degree polynomial as follows. Suppose we had a good degree-d univariate approximation to the sign function $p_d(x)\approx\text{sgn}(x)$, and say we have an n-dimensional halfspace $\text{sgn}(v\cdot x-\theta)$. Then, $\text{sgn}(v\cdot x-\theta)\approx p_d(v\cdot x-\theta)$. Moreover, this latter quantity is now a degree-d multivariate polynomial. The sense in which we measure approximations will be distributional, the L_2 error of our multivariate polynomial over the distribution $\mathcal{D}$. Hence, we need a polynomial p_d that well-approximates the sign function on the marginal distribution in the direction v, i.e., the distribution over projections onto the vector v.

For the uniform distribution on a sphere, the projection onto a single coordinate is distributed very close to Gaussian distribution. For a log-concave distribution, its projection is distributed log-concavely. In both of these cases, it so happens that the necessary degree to get approximation error ϵ boils down to a one-dimensional problem! For the sphere, we can upper-bound the degree necessary as a function of ϵ using the following for the normal distribution $N(0,\frac{1}{\sqrt{2}})$ with density $e^{-x^2}/\sqrt{\pi}$:

Theorem 6 *For any $d > 0$ and any $\theta \in \mathbb{R}$, there is a degree-d univariate polynomial $p_{d,\theta}$ such that*

$$\int_{-\infty}^{\infty} (p_{d,\theta}(x) - \mathrm{sgn}(x-\theta))^2 \frac{e^{-x^2}}{\sqrt{\pi}} dx = O\left(\frac{1}{\sqrt{d}}\right). \tag{5}$$

We note that the $n^{O(1/\epsilon^2)}$-time, $O(\mathsf{opt} + \epsilon)$-error analogues of Theorem 1, part 1, mentioned in Section 1.3 follows from Fact 5 and Theorem 6 using the L_2-norm analogue of the polynomial regression algorithm mentioned at the end of Section 3. The improved time bound comes from the fact that we no longer need to invoke $\mathbf{E}[|Z|] \leq \sqrt{\mathbf{E}[Z^2]}$ to bound the square loss, since we are minimizing the square loss directly rather than the absolute loss.

Proof of Theorem 6: We assume without loss of generality that $\theta \geq 0$; an entirely similar proof works for $\theta < 0$. First, suppose that $\theta > \sqrt{d}$. Then we claim that the constant polynomial $p(x) = -1$ will be a sufficiently good approximation of $\mathrm{sgn}(x-\theta)$. In particular, it will have error,

$$\int_{\theta}^{\infty} \frac{4e^{-x^2}}{\sqrt{\pi}} dx \leq \int_{\sqrt{d}}^{\infty} \frac{4e^{-x}}{\sqrt{\pi}} dx = \frac{4e^{-\sqrt{d}}}{\sqrt{\pi}} \leq \frac{4}{\sqrt{\pi d}}.$$

So the case that $\theta > \sqrt{d}$ is easy, and for the remainder we assume that $\theta \in [0, \sqrt{d}]$.

We use the Hermite Polynomials H_d, $d = 0, 1, \ldots$, (H_d is a degree-d *univariate* polynomial) which are a set of orthogonal polynomials given the weighting $e^{-x^2}\pi^{-1/2}$. In particular,

$$\int_{-\infty}^{\infty} H_{d_1}(x) H_{d_2}(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx = \begin{cases} 0 & \text{if } d_1 \neq d_2 \\ 2^{d_1} d_1! & \text{if } d_1 = d_2 \end{cases}$$

Hence these polynomials form an orthogonal basis of polynomials with respect to the inner product $\langle p, q \rangle = \int_{-\infty}^{\infty} p(x)q(x)e^{-x^2}\pi^{-1/2} dx$. The functions $\bar{H}_d(x) = H_d(x)/\sqrt{2^d d!}$ are an orthonormal basis.

Now, the best degree d approximation to the function $\mathrm{sgn}(x-\theta)$, in the sense of (5), for any d, can be written as $\sum_{i=0}^{d} c_i \bar{H}_i(x)$. The $c_i \in \mathbb{R}$ that minimize (5) are,

$$\begin{aligned} c_i &= \int_{-\infty}^{\infty} \mathrm{sgn}(x-\theta) \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \\ &= \int_{\theta}^{\infty} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx - \int_{-\infty}^{\theta} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \\ &= 2\int_{\theta}^{\infty} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \quad \text{(for } i \geq 1\text{)} \end{aligned} \tag{6}$$

The last step follows from the fact that $\int_{-\infty}^{\infty} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx = 0$ for $i \geq 1$ by orthogonality of $\bar{H}_i$ with $\bar{H}_0$. Next, to calculate our error, we use Parseval's identity,

$$\int_{-\infty}^{\infty} \left(\sum_{i=0}^{d} c_i \bar{H}_i(x) - \mathrm{sgn}(x-\theta)\right)^2 \frac{e^{-x^2}}{\sqrt{\pi}} dx = 1 - \sum_{i=0}^{d} c_i^2 = \sum_{i=d+1}^{\infty} c_i^2.$$

The above holds because $\int_{-\infty}^{\infty} \frac{e^{-x^2}}{\sqrt{\pi}} = 1$ and hence $\sum_{i=0}^{\infty} c_i^2 = 1$ ($\mathrm{sgn}(x) \in L^2(\mathbb{R}, e^{-x^2})$ and polynomials are dense in this set). It thus suffices for us to bound $\sum_{i=d+1}^{\infty} c_i^2$.

It is now easy to calculate each coefficient c_i using standard properties of the Hermite Polynomials. It is well known [35] that the Hermite polynomials can be defined by:

$$H_i(x)e^{-x^2} = (-1)^i \frac{d^n}{dx^n} e^{-x^2}, \text{ which implies } \frac{d}{dx} H_i(x)e^{-x^2} = -H_{i+1}(x)e^{-x^2}.$$

In turn, this and (6) imply that for $i \geq 1$,

$$\begin{aligned} c_i &= \frac{2}{\sqrt{\pi 2^i i!}} \int_\theta^\infty H_i(x) e^{-x^2} dx \\ &= \frac{2}{\sqrt{\pi 2^i i!}} \left(-H_{i-1}(x) e^{-x^2} \right) \Big|_\theta^\infty \\ &= \frac{2}{\sqrt{\pi 2^i i!}} H_{i-1}(\theta) e^{-\theta^2}. \end{aligned} \tag{7}$$

We must show that $\sum_{i=d+1}^\infty c_i^2 = O(1/\sqrt{d})$. To do this, it suffices to show that for each i we have $c_i^2 = O(i^{-3/2})$. From (7) we have, for $i \geq 1$,

$$c_i^2 = \frac{4}{\pi 2^i i!} (H_{i-1}(\theta))^2 e^{-2\theta^2}. \tag{8}$$

Now, conveniently Theorem 1.i of [7] states that, for all $i \geq \theta^2$,

$$\frac{1}{2^i i!} H_i(\theta)^2 e^{-\theta^2} \leq \frac{C}{\sqrt{i}}$$

where C is some absolute constant. Since we have $\theta \leq \sqrt{d}$ by assumption, we have that for $i \geq d+1$, $c_i^2 \leq \frac{4C}{2\pi i \sqrt{i-1}}$, which is of the desired form $O(i^{-3/2})$, and Theorem 6 is proved. ■

With Theorem 6 in hand it is not difficult to to establish Theorem 1 Part 1(b), which we restate below:

> Let $\mathcal{D}$ be a distribution over $\mathbb{R}^n \times \{-1, 1\}$, with $\mathcal{D}_X$ uniform over S^{n-1}. With probability $1 - \delta$, the L_1 polynomial regression outputs a hypothesis with error $\mathsf{opt} + \epsilon$ given $\mathrm{poly}(n^{1/\epsilon^4}, \log \frac{1}{\delta})$ examples.

Proof: Let $f(x) = \mathrm{sgn}(v \cdot x - \tau)$ be any halfspace over the unit ball S^{n-1}, where without loss of generality we may assume $\|v\| = 1$ (and thus $|\tau| \leq 1$). Let $\mathcal{U}$ denote the uniform distribution over S^{n-1}. It suffices to establish the existence of a degree-d polynomial $P(x)$, with $d = O(1/\epsilon^4)$, which satisfies the condition $\mathbf{E}_{x \in \mathcal{U}}[(P(x) - f(x))^2] \leq \epsilon^2$; given such a polynomial we apply Theorem 5 and Theorem 1 Part 1(b) immediately follows.

Let $\theta = \sqrt{\frac{n-3}{2}}\tau$ and let $P(x) = p_{d,\theta}(\sqrt{\frac{n-3}{2}} v \cdot x)$. For $d = O(1/\epsilon^4)$, we show that the polynomial $P(x) = p_{d,\theta}\left(\sqrt{\frac{n-3}{2}}(v \cdot x)\right)$ satisfies $\mathbf{E}_{\mathcal{U}}[(P(x) - f(x))^2] \leq \epsilon^2$.

We have (justifications are given below):

$$\begin{aligned} \mathbf{E}_{x \in \mathcal{U}}[(P(x) - f(x))^2] &= \mathbf{E}_{x \in \mathcal{U}}\left[\left(p_{d,\theta}\left(\sqrt{\tfrac{n-3}{2}}(v \cdot x)\right) - \mathrm{sgn}\left(\sqrt{\tfrac{n-3}{2}}(v \cdot x) - \theta\right)\right)^2\right] \\ &= \frac{A_{n-2}}{A_{n-1}} \int_{-1}^{1} (1 - z^2)^{(n-3)/2} \left(p_{d,\theta}\left(\sqrt{\tfrac{n-3}{2}} z\right) - \mathrm{sgn}(\sqrt{\tfrac{n-3}{2}} z - \theta)\right)^2 dz \qquad (9) \\ &\leq \frac{A_{n-2}}{A_{n-1}} \int_{-\infty}^{\infty} e^{-z^2(n-3)/2} \left(p_{d,\theta}\left(\sqrt{\tfrac{n-3}{2}} z\right) - \mathrm{sgn}(\sqrt{\tfrac{n-3}{2}} z - \theta)\right)^2 dz \qquad (10) \\ &= \frac{A_{n-2}}{A_{n-1}} \int_{-\infty}^{\infty} e^{-y^2} \left(p_{d,\theta}(y) - \mathrm{sgn}(y - \theta)\right)^2 \frac{dy}{\sqrt{(n-3)/2}} \qquad (11) \\ &\leq \epsilon^2 \qquad (12) \end{aligned}$$

where (9) follows from Fact 10 on the pdf of the uniform distribution over S^{n-1}; (10) follows from $1 - z \leq \exp(-z)$ and the fact that the integrand is nonnegative; (11) follows from a change of variable $y = \sqrt{\frac{n-3}{2}} \cdot z$; and (12) follows from $\frac{A_{n-2}}{A_{n-1}} = \Theta(\sqrt{n})$, Theorem 6, and our choice of $d = O(1/\epsilon^4)$. This concludes the proof of Theorem 1 Part 1(b). ■

Since we have proven Theorem 1 Part 1(a) in Section 4, we are now ready to move on to the log-concave part. The first thing to notice is that, just as the normal distribution served as a prototypical distribution for all spheres, there is a log-concave distribution that is not much smaller than any other:

Lemma 6 *Let ν be the distribution on $\mathbb{R}$ with density $d\nu(x) = e^{-|x|/16}/32$. Let μ be any log-concave distribution on $\mathbb{R}$ with mean 0 and variance 1. Then, for all $x \in \mathbb{R}$, $d\mu(x) \leq (32e)d\nu(x)$.*

In the above, we necessarily chose a distribution ν that did not have variance 1.

Proof: To prove this lemma, we will use the properties of log-concave functions given by Lovasz and Vempala [29]. Specifically, for any log-concave density $d\mu$ with mean 0 and variance 1, $\forall x\ d\mu(x) \leq 1$, and $d\mu(0) \geq 1/8$. From the latter fact, we next argue that $d\mu(x) \leq e^{-|x|/16}$ for $|x| > 16$. It suffices to show this for $x > 16$ by symmetry. Suppose not, i.e., suppose $\exists r > 16\ d\mu(r) \geq e^{-r/16}$. Then log-concavity implies that $d\mu(x) \geq (1/8)^{1-x/r}(e^{-r/16})^{x/r}$ for $x \in [0, r]$. In turn, this means,

$$\int_0^{16} d\mu(x) \geq \int_0^{16} \frac{1}{8} e^{-x/16} dx > 1,$$

which is a contradiction. Hence, $d\mu(x) \leq e^{-|x|/16} = 32 d\nu(x)$ for $|x| > 16$. (These bounds are far from tight.) Also, for $|x| < 16$, $d\mu(x) \leq 1 \leq (32e)d\nu(x)$. ■

This lemma will enable us to transfer a bound on the error of a *fixed* log-concave function such as $e^{-2|x|}$ to *all* log-concave functions.

Lemma 7 *There exists a fixed function $d : \mathbb{R} \to \mathbb{R}$, such that, for any log-concave distribution μ, and any $\theta \in \mathbb{R}$, there exists a degree-$d(\epsilon)$ polynomial p, such that*

$$\int_{-\infty}^{\infty} (p(x) - \operatorname{sgn}(x - \theta))^2 d\mu(x) \leq \epsilon.$$

Proof: It suffices to show it for any log-concave distribution μ with mean 0 and variance 1. This is because we can always apply an affine transformation to x, $x \to ax + b$ which puts it in such standard position and maintains the properties of the lemma (for a suitably transformed polynomial p and θ). Thus, we assume that μ has mean 0 and variance 1.

Next, we claim it suffices to show the lemma for the log-concave density $d\nu(x) = e^{-|x|/16}/32$, which has mean 0 but variance > 1. To see this, suppose it holds for $d\nu$ and p, and we have some mean 0 variance 1 log-concave density $d\mu$. Then by Lemma 6,

$$\int_{-\infty}^{\infty} (p(x) - \operatorname{sgn}(x - \theta))^2 d\mu(x) \leq 32e \int_{-\infty}^{\infty} (p(x) - \operatorname{sgn}(x - \theta))^2 d\nu(x) \leq 32e\epsilon.$$

Hence it would hold for mean-0 variance-1 $d\mu$ with function $d' : \mathbb{R} \to \mathbb{R}$ where $d'(\epsilon) = d(\epsilon/(32e))$. By a similar stretching argument, it suffices to show it for $d\nu(x) = e^{-2|x|}$.

Next, again WLOG, it suffices to show it for $|\theta| < \log 1/\epsilon$. For if $|\theta| > 2 \log 1/\epsilon$, then the constant polynomial $p(x) = -\operatorname{sgn}(\theta)$ has error less than ϵ under $d\nu(x) = e^{-2|x|}$. Continuing on the seemingly endless chain of WLOGs, we next that it suffices to show it for $\theta = 0$. Suppose it holds for $d\nu(x) = e^{-2|x|}$, a particular p and ϵ, and $\mathrm{sgn}(x)$. That is,

$$\int_{-\infty}^{\infty} (p(x) - \mathrm{sgn}(x))^2 d\nu(x) \leq \epsilon \tag{13}$$

Then consider the function $\mathrm{sgn}(x-\theta)$ and the density $d\rho(x) = e^{-2|x-\theta|/\log(1/\epsilon)}/\log(1/\epsilon)$. For this density, by (13) and change of variable $z = \log(1/\epsilon)(x-\theta)$,

$$\int_{-\infty}^{\infty} (p(z) - \mathrm{sgn}(z))^2 d\nu(z) = \int_{-\infty}^{\infty} (p(\log(1/\epsilon)(x-\theta)) - \mathrm{sgn}(x-\theta))^2 d\rho(x) \leq \epsilon \tag{14}$$

Now, observe that as long as $\log(1/\epsilon) > 1$, $(\epsilon \leq 1/e)$

$$\frac{d\nu(x)}{d\rho(x)} = \log(1/\epsilon) e^{2(\frac{|x-\theta|}{\log(1/\epsilon)} - |x|)} \leq \log(1/\epsilon) e^{2(\frac{|x-\theta| - |x|}{\log(1/\epsilon)})} \leq \log(1/\epsilon) e^{2\frac{|\theta|}{\log(1/\epsilon)}} \leq \log(1/\epsilon)e^2.$$

By this and (14),

$$\int_{-\infty}^{\infty} (p(\log(1/\epsilon)(x-\theta)) - \mathrm{sgn}(x-\theta))^2 d\mu(x) \leq e^2 \epsilon \log(1/\epsilon).$$

Hence a bound of ϵ on the error of p for $\mathrm{sgn}(x)$ implies a bound of $e^2\epsilon\log(1/\epsilon)$ on the error of $p(\log(1/\epsilon)(x-\theta))$. So, it suffices to show we can achieve such a bound for $\mathrm{sgn}(x)$, $d\nu(x) = e^{-2|x|}$, and arbitrarily small ϵ.

At this point we have a single function $\mathrm{sgn}(x)$, a single density $e^{-2|x|}$, and we must establish that for any ϵ there is some $d = d(\epsilon)$ for which there is a degree-d polynomial p for which (13) holds. But $\mathrm{sgn}(x) \in L^2(\mathbb{R}, e^{-2|x|})$ because $\int_{-\infty}^{\infty} \mathrm{sgn}(x)^2 e^{-2|x|} dx = 1 < \infty$ and it is known that polynomials are dense in $L^2(\mathbb{R}, e^{-2|x|})$ [35]. ■

4.1 Agnostically Learning Disjunctions under Any Distribution.

We can use the polynomial regression algorithm to learn disjunctions agnostically with respect to any distribution in subexponential time. We make use of the existence of low-degree polynomials which strongly approximate the OR function in the L_∞ norm:

Theorem 7 *[33, 31, 22] Let $f(x_1, \ldots, x_n)$ compute the OR function on some subset of (possibly negated) input variables. Then there exists a polynomial p of degree $O(\sqrt{n}\log(1/\epsilon))$ such that for all $x \in \{-1,1\}^n$, we have $|f(x) - p(x)| \leq \epsilon$.*

For $\epsilon = \Theta(1)$ this fact appears in [33, 31]; an easy extension to arbitrary ϵ is given in [22]. Theorem 2 follows immediately from Theorems 7 and Theorem 5, since for any distribution $\mathcal{D}$ the L_∞ bound given by Theorem 7 clearly implies the bound on expectation required by Theorem 5.

We note that low-degree L_∞-approximators are know for richer concept classes than just disjunctions. For example, results of O'Donnell and Servedio [32] show that any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ computed by a Boolean formula of linear size and constant depth is ϵ-approximated in the L_∞ norm by a polynomial of degree $\tilde{O}(\sqrt{n}) \cdot \mathrm{poly}\log\frac{1}{\epsilon}$. By combining Theorem 5 with such existence results, one can immediately obtain arbitrary-distribution agnostic learning results analogous to Theorem 2 for those concept classes as well.

4.2 Hardness results for agnostically learning halfspaces over the hypercube

In this section we show that the challenging "learning noisy parity" problem reduces to the problem of agnostically learning halfspaces with respect to the uniform distribution over the hypercube. Recall that a vector $c \in \{0,1\}^n$ induces a parity function $c : \{0,1\}^n \to \{0,1\}$ as follows: $c(x) = c \cdot x \bmod 2$ (the indices of c equal to 1 are the *relevant* variables). The *noisy parity learning problem* is the problem of PAC learning an unknown parity function with respect to the uniform distribution on $\{0,1\}^n$ where the *label* of each example is flipped (independently) with probability η. The fastest known learning algorithm for this well-known problem is due to Blum *et al.* [5] and runs in time $2^{O(n/\log n)}$.

An algorithm for agnostically learning halfspaces can be easily transformed into an algorithm for learning parity with noise:

Theorem 8 *Let A be an algorithm for agnostically learning halfspaces to accuracy $\mathsf{opt} + \epsilon$ with respect to the uniform distribution over $\{0,1\}^n$ running in time $t = t(1/\epsilon, n)$. Then there exists an algorithm B for learning parity with noise which runs in time* $\mathrm{poly}(n,t)$.

Proof: Assume that the unknown parity function c has k relevant variables (and for simplicity assume k is even). Note that for a set S of k variables, the majority function on S (equal to 1 if $\lfloor k/2 \rfloor + 1$ or more of the variables in S are set to 1) agrees with the parity function on all variables in S for a $1/2 + 1/\sqrt{k}$ fraction of inputs of $\{0,1\}^n$. This is because the majority function equals parity for all inputs of hamming weight equal to $k/2$ and agrees with parity on half of all other inputs.

Now choose a random example (labeled by c) and flip its label with probability η. The probability that the majority function on S correctly labels the example equals $\eta + (1-2\eta)(1/2 - 1/\sqrt{k}) = 1/2 - (1-2\eta)/\sqrt{k}$. That is, the *error rate* of the majority function on S with respect to noisy examples is bounded away from $1/2$ by $(1-2\eta)/\sqrt{k}$.

We can now use an algorithm for agnostically learning halfspaces to identify the relevant variables of the unknown parity function c. To determine if the variable x_i is relevant, set $\epsilon = (1/2)(1-2\eta)/\sqrt{k}$ and take a number of random examples as specified by the agnostic learner. Feed the examples to the agnostic learner *with the ith bit removed from every example.* If x_i is a relevant variable, then the labels will be totally uncorrelated with the examples (now of length $n-1$), and the agnostic learner will not produce a hypothesis with error rate bounded away from $1/2$. If x_i is irrelevant, then the majority function on the relevant variables has error rate bounded away from $1/2$, and the agnostic learner will output a hypothesis with error less than $1/2 - (1/2)(1-2\eta)/\sqrt{k}$. ■

If the error rate η is $\Theta(1)$ and the agnostic learning algorthm runs in time $n^{O(1/\epsilon^{2-\beta})}$, then the above algorithm will learn a noisy parity in time $2^{O(n^\gamma)}$ for some $0 < \gamma < 1$.

4.3 An application to learning intersections of halfspaces

Learning an intersection of halfspaces is a challenging and well-studied problem even in the noise-free setting. Klivans *et al.* [22] showed that the standard low-degree algorithm can learn the intersection of k halfspaces with respect to the uniform distribution on $\{-1,1\}^n$ to error ϵ in time $n^{O(k^2/\epsilon^2)}$, provided that $\epsilon < 1/k^2$. Note that because of the requirement on ϵ, the algorithm always takes time at least $n^{\Omega(k^6)}$ even if the desired final error is $\epsilon = \Theta(1)$ independent of k.

We can use the idea of learning halfspaces agnostically to obtain the following runtime bound which is better than [22] for $\epsilon > \frac{1}{k}$:

Theorem 9 *Let $f = h_1 \wedge \ldots \wedge h_k$ be an intersection of k halfspaces over $\{-1,1\}^n$. Then f is learnable with respect to the uniform distribution over $\{-1,1\}^n$ in time $n^{O(k^4/\epsilon^2)}$ for any $\epsilon > 0$.*

We note that a comparable bound can be proved via techniques from recent work due to Jackson et al. [18] which does not involve agnostic learning. The presentation here, however, is more straightforward and shows how agnostic learning can have applications even in the non-noisy framework.

The approach that establishes Theorem 9 is similar to Jackson's Harmonic Sieve [17]: we apply a boosting algorithm, using the polynomial regression algorithm at each stage to identify a low-degree polynomial which, after thresholding, has advantage at least $\Omega(1/k)$ on the target function.

We begin with the following easy fact which follows directly from the "discriminator lemma" [15]:

Fact 8 *Let $f = h_1 \wedge \ldots \wedge h_k$ be an intersection of k halfspaces. Then for any distribution $\mathcal{D}$ on $\{0,1\}^n$ either there exists an h_i such that $|E_{\mathcal{D}}[fh_i]| \geq 1/k$ or we have $|E_{\mathcal{D}}[f]| \geq 1/k$.*

Hence for any distribution $\mathcal{D}$ there exists a single halfspace which has accuracy at least $1/2+1/2k$ with respect to f and $\mathcal{D}$. We will be concerned only with distributions that are c-bounded (c will be chosen later), i.e. distributions D such that $D(x) \leq c/2^n$ for all x. Fix such a c-bounded distribution $\mathcal{D}$ and let $h_{\mathcal{D}}$ denote the halfspace obtained from Fact 8. Applying Fact 5 it is not difficult to see that for any halfspace (and in particular $h_{\mathcal{D}}$) and sufficiently large constant a,

$$\sum_{S, |S| \geq a \cdot k^4 c^2} \hat{h_{\mathcal{D}}}(S)^2 \leq c/16k^2.$$

By setting $g = \sum_{S,|S| \leq a \cdot k^4 c^2} \hat{h_{\mathcal{D}}}(S) \chi_S(x)$, we have $E_{\mathcal{D}}[|g - h_D|] \leq 1/4k$ for any c-bounded distribution $\mathcal{D}$.

We now show that the polynomial regression algorithm can be used as a weak learning algorithm for f:

Lemma 9 *There exists an algorithm A such that for any c-bounded distribution $\mathcal{D}$ and $0 < \delta < 1$, if A is given access to examples drawn from $\mathcal{D}$ labeled according to f, then A runs in time $poly(n^{k^4 c^2}, 1/\delta)$ and with probability at least $1-\delta$, A outputs a hypothesis h such that $Pr_{\mathcal{D}}[f(x) = h(x)] \geq 1/2 + 1/8k$.*

Proof: Let $\ell = ak^4c^2$ for a sufficiently large constant a. Apply the polynomial regression algorithm from Section 3 to obtain a hypothesis $g^* = \mathrm{sgn}\left(\sum_{|S| \leq \ell} w_S \chi_S(x) - t\right)$. For $\xi > 0$, we claim that g^* has error less than $1/2 - 1/4k + \xi$ as long as $m \geq \mathrm{poly}(n^\ell, 1/\xi^2, \log(1/\delta))$ as in Theorem 5. To see this note that

$$E_{\mathcal{D}}[|f(x) - g^*|] \leq E_{\mathcal{D}}[|f(x) - h_{\mathcal{D}}(x)|] + E_{\mathcal{D}}[|h_{\mathcal{D}}(x) - g^*|]$$

and recall that the first term on the right hand side is at most $1/2 - 1/2k$. For the second term, recall that $\min_w E_{\mathcal{D}}[|h_{\mathcal{D}}(x) - \sum_{|S| \leq \ell} w_S \chi_S|] \leq 1/4k$. But g^* is an approximation to the truncated Fourier polynomial for $h_{\mathcal{D}}(x)$ and as in the proof of Theorem 5, for our choice of m, $E_{\mathcal{D}}[|h_{\mathcal{D}}(x) - g^*(x)|] \leq \min_w E_{\mathcal{D}}[|h_{\mathcal{D}}(x) - \sum_{|S| \leq \ell} w_S \chi_S|] + \xi$ with probability greater than $1-\delta$. Hence with probability $1-\delta$ we have $E_{\mathcal{D}}[|f(x) - g^*(x)|] \leq 1/2 - 1/4k + \xi$. Taking $\xi = 1/(8k)$ gives the lemma. ■

At this point we will need to recall the definition of a *boosting* algorithm, see e.g. [13]. Roughly speaking, a boosting algorithm iteratively applies a weak learning algorithm as a subroutine in order to construct a highly accurate final hypothesis. At each iteration, the boosting algorithm generates a distribution $\mathcal{D}$ and runs the weak learner to obtain a hypothesis which has accuracy $1/2 + \gamma$ with respect to $\mathcal{D}$. After $t = \mathrm{poly}(1/\gamma, 1/\epsilon)$ iterations, the boosting algorithm outputs a hypothesis with accuracy greater than $1 - \epsilon$. The following fact from [23] is sufficient for our purposes:

Theorem 10 *There is a boosting algorithm which runs in $t = O(1/\epsilon^2\gamma^2)$ iterations and at each stage generates an $O(1/\epsilon)$-bounded distribution $\mathcal{D}$.*

By combining this boosting algorithm with the weak learning algorithm from Lemma 9 we obtain Theorem 9:

Proof of Theorem 9: Run the boosting algorithm to learn f using the weak learner from Lemma 9 as a subroutine. The boosting algorithm requires at most $O(1/\epsilon^2 k^2)$ iterations since the distributions are all $O(1/\epsilon)$ bounded and the weak learner outputs a hypothesis with accuracy $1/2+\Omega(1/k)$. The running time of the weak learning algorithm is at most $n^{O(k^4/\epsilon^2)}$ since each distribution is $c = O(1/\epsilon)$ bounded. ∎

5 Learning halfspaces over the sphere with the degree-1 version of the polynomial regression algorithm

Let us return to the case, where the marginal distribution $\mathcal{D}_X$ is uniform over S^{n-1}, and now consider the homogeneous $d = 1$ version of the polynomial regression algorithm. In this case, we would like to find the vector $w \in \mathbb{R}^n$ that minimizes $\mathbf{E}_{\mathcal{D}_X}[(w \cdot x - y)^2]$. By differentiating with respect to w_i and using the fact that $\mathbf{E}[x_i] = \mathbf{E}[x_i x_j] = 0$ for $i \neq j$ and $\mathbf{E}[x_i^2] = \frac{1}{n}$, we see that the minimum is achieved at $w_i = \frac{1}{n}E[x_i y_i]$.

This is essentially the same as the simple `Average` algorithm which was proposed by Servedio in [34] for learning origin-centered halfspaces under uniform in the presence of random misclassification noise. The `Average` algorithm draws examples until it has a sample of m positively labeled examples $x^1, \ldots, x^m$, and then it returns the hypothesis $h(x) = \mathrm{sgn}(\overline{v} \cdot x)$ where $\overline{v} = \frac{1}{m}\sum_{i=1}^m x^i$ is the vector average of the positive examples. The intuition for this algorithm is simple: if there were no noise then the average of the positive examples should (in the limit) point exactly in the direction of the target normal vector.

A straightforward application of the bounds from Section 3 and Section 4 implies only that the degree-1 polynomial regression algorithm should achieve some fixed constant accuracy $\Theta(1)$ independent of opt for agnostic learning halfspaces under the uniform distribution on S^{n-1}. However, a more detailed analysis shows that the simple `Average` algorithm does surprisingly well, in fact obtaining a hypothesis with error rate $O(\mathsf{opt}\sqrt{\log(1/\mathsf{opt})}) + \epsilon$; this is Theorem 3. We give useful preliminaries in Section 5.1 and prove Theorem 3 in Section 5.2.

5.1 Learning Halfspaces on the Unit Sphere: Preliminaries

We write S^{n-1} to denote the n-dimensional Euclidean sphere $S^{n-1} = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i^2 = 1\}$. Given two nonzero vectors $u, v \in \mathbb{R}^n$ we write $\alpha(u, v)$ to denote $\arccos(\frac{u \cdot v}{\|u\| \cdot \|v\|})$, the angle between u and v. If the target halfspace is $\mathrm{sgn}(u \cdot x)$ and $\mathrm{sgn}(v \cdot x)$ is a hypothesis halfspace, then it is easy to see that we have $\Pr_{x \in \mathcal{U}}[\mathrm{sgn}(u \cdot x) \neq \mathrm{sgn}(v \cdot x)] = \alpha(u, v)/\pi$.

We write A_{n-1} to denote the surface area of S^{n-1}. It is well known (see e.g. [1]) that $A_{n-2}/A_{n-1} = \Theta(n^{1/2})$. The following fact (see e.g. [1]) is useful:

Fact 10 *For any unit vector $v \in \mathbb{R}^n$ and any $-1 \le \alpha < \beta \le 1$, we have*

$$\Pr_{x \in \mathcal{U}}[\alpha \le v \cdot x \le \beta] = \frac{A_{n-2}}{A_{n-1}} \cdot \int_\alpha^\beta (1 - z^2)^{(n-3)/2} dz.$$

The following straightforward result lets us deal easily with sample error:

Fact 11 *Let $\mathcal{D}$ be any distribution over S^{n-1}. Let v denote the expected location $\mathbf{E}_{x \in \mathcal{D}}[x]$ of a random draw from $\mathcal{D}$, and suppose that $\|v\| \ge \xi$. Then if $\overline{v} = \frac{1}{m}\sum_{i=1}^m x^i$ is a sample estimate of $\mathbf{E}_{x \in \mathcal{D}}[x]$ where each x^i is drawn independently from $\mathcal{D}$ and $m = O(\frac{n}{\epsilon^2 \xi^2} \log \frac{n}{\delta})$, we have that $\Pr_{x \in \mathcal{U}}[sgn(\overline{v} \cdot x) \ne \mathrm{sgn}(v \cdot x)] \le \epsilon$ with probability at least $1 - \delta$.*

Proof: We define an orthonormal basis for $\mathbb{R}^n$ by letting vector u^1 denote $\frac{v}{\|v\|}$ and letting $u^2, \ldots, u^n$ be an arbitrary orthonormal completion. Given a vector $z \in \mathbb{R}^n$, we may write z_1 for $z \cdot u^1$ and $z_2, \ldots, z_n$ for $z \cdot u^2, \ldots, z \cdot u^n$ respectively. We have $\mathbf{E}_{x \in \mathcal{D}}[x_1] = \xi$ so standard additive Chernoff bounds imply that taking $m = O(\frac{1}{\xi^2} \log \frac{1}{\delta})$ many draws will result in $|\overline{v}_1 - \xi| \le \frac{\xi}{2}$ with probability at least $1 - \frac{\delta}{2}$. For $i = 2, \ldots, n$ we have $\mathbf{E}_{x \in \mathcal{D}}[x_i] = 0$; again standard additive Chernoff bounds imply that taking $m = O(\frac{n}{\epsilon^2 \xi^2} \log \frac{n}{\delta})$ many draws will result in $|\overline{v}_i| \le \frac{\epsilon \xi}{2\sqrt{n}}$ for each i with probability at least $1 - \frac{\delta}{2}$. Thus, with overall probability at least $1 - \delta$ we have

$$\alpha(\overline{v}, v) = \arctan\left(\frac{\sqrt{\overline{v}_2^2 + \cdots + \overline{v}_n^2}}{\overline{v}_1}\right) \le \arctan(\epsilon) \le \epsilon$$

and thus $\Pr_{x \in \mathcal{U}}[sgn(\overline{v} \cdot x) \ne \mathrm{sgn}(v \cdot x)] \le \alpha(\overline{v}, v)/\pi < \epsilon/\pi < \epsilon$. ■

5.2 Proof of Theorem 3

We have that $\mathcal{D}$ is a distribution over $X \times \{-1, 1\}$ whose marginal is the uniform distribution $\mathcal{U}$ on S^{n-1}. Without loss of generality we may suppose that the optimal origin-centered halfspace is $f(x) = \mathrm{sgn}(x_1)$, i.e. the normal vector to the separating hyperplane is $e_1 = (1, 0, \ldots, 0)$. We write S^+ to denote the "positive hemisphere" $\{x \in S^{n-1} : x_1 \ge 0\}$ and write S^- to denote $S^{n-1} \setminus S^+$. We may also suppose without loss of generality that the optimal halfspace's error rate opt is such that $O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}})$ is less than $\frac{1}{4}$, i.e. opt is less than some fixed absolute constant that we do not specify here.

Let $p : S^{n-1} \to [0, 1]$ be the function

$$p(z) = \Pr_{(x,y) \in \mathcal{D}}[y \ne f(z) \mid x = z] \tag{15}$$

so intuitively $p(z)$ is the probability of getting a "noisy label" y on instance z. (We assume the joint distribution $\mathcal{D}$ on $X \times Y$ is sufficiently "nice" in terms of measurability, etc. so that p is well-defined as specified above.) Let v denote the true vector average of all positively labeled examples generated by $\mathcal{D}$, i.e.

$$v = \int_{x \in S^+} x(1 - p(x))\mathcal{U}(x) \ + \int_{x \in S^-} xp(x))\mathcal{U}(x).$$

If the number m of examples used by `Average` went to infinity, the vector average $\overline{v}$ that `Average` computes would converge to v. We prove Theorem 3 by first establishing bounds on v, and then using Fact 11 (in Appendix 5.1) to deal with sample error.

Let u denote the vector average of all points in S^+. It is clear from symmetry that $u = (u_1, 0, \ldots, 0)$ for some $u_1 > 0$; in fact we have

Claim 12 $u_1 = 2 \cdot \frac{A_{n-2}}{A_{n-1}} \cdot \int_0^1 z(1-z^2)^{(n-3)/2} dz = \Theta(\frac{1}{\sqrt{n}})$.

Proof: The first equality follows immediately from Fact 10 (the factor of 2 is present because u is the vector average of half the points of S^{n-1}). For the second equality, since $\frac{A_{n-2}}{A_{n-1}} = \Theta(\sqrt{n})$ we need to show that $\int_0^1 z(1-z^2)^{(n-3)/2} dz$ is $\Theta(1/n)$. For each $z \in [1/\sqrt{n}, 2/\sqrt{n}]$ the value of the integrand $z(1-z^2)^{(n-3)/2}$ is at least $(1/\sqrt{n})(1-\frac{4}{n})^{(n-3)/2} = \Theta(1/\sqrt{n})$, so this implies that the whole integral is $\Omega(1/n)$. The integrand is clearly at most $1/\sqrt{n}$ for all $z \in [0, 1/\sqrt{n}]$, so we have $\int_0^{2/\sqrt{n}} z(1-z^2)^{(n-3)/2} dz = \Theta(1/n)$; to finish the proof we need only show that $\int_{2/\sqrt{n}}^1 z(1-z^2)^{(n-3)/2} dz = O(1/n)$. We can piecewise approximate this integral (in increments of $1/\sqrt{n}$) as

$$\int_{2/\sqrt{n}}^1 z(1-z^2)^{(n-3)/2} dz \approx \sum_{j=2}^{\sqrt{n}} \frac{j}{\sqrt{n}} e^{-j^2/2} \cdot \frac{1}{\sqrt{n}} = \frac{1}{n} \sum_{j=2}^{\sqrt{n}} j e^{-j^2/2} < \frac{1}{n} \sum_{j=2}^{\infty} j e^{-j^2/2} = O(1/n)$$

and this gives the claim. ■

If there were no noise then then the vector average v would equal u; since there is noise we must add in a contribution from true negative examples that are falsely labeled as positive, and subtract off a contribution from true positive examples that are falsely labeled as negative.

Let opt_{-} and opt_{+} be defined as

$$\mathsf{opt}_{-} = \int_{x \in S^-} p(x)\mathcal{U}(x) \qquad \text{and} \qquad \mathsf{opt}_{+} = \int_{x \in S^+} p(x)\mathcal{U}(x),$$

so opt_{-} is the overall probability of receiving an example that is truly negative but falsely labeled as positive, and vice versa for opt_{+}. Clearly $\mathsf{opt} = \mathsf{opt}_{-} + \mathsf{opt}_{+}$. Let u^- and u^+ be the vectors

$$u^- = \frac{\int_{x \in S^-} x p(x)\mathcal{U}(x)}{\mathsf{opt}_{-}} \qquad \text{and} \qquad u^+ = \frac{\int_{x \in S^+} x p(x)\mathcal{U}(x)}{\mathsf{opt}_{+}}$$

so u^- (u^+ respectively) is the vector average of all the false positive (false negative respectively) examples generated by p. Then the vector average v of all positively labeled examples is

$$v = \frac{u/2 + \mathsf{opt}_{-} u^- - \mathsf{opt}_{+} u^+}{1/2 + \mathsf{opt}_{-} - \mathsf{opt}_{+}} = C_1 \cdot v'$$

where $v' = u/2 + \mathsf{opt}_{-} u^- - \mathsf{opt}_{+} u^+$ and $\frac{4}{3} \leq C_1 = \frac{1}{1/2 + \mathsf{opt}_{-} - \mathsf{opt}_{+}} \leq 4$; the bounds on C_1 hold since by assumption we have $\mathsf{opt} \leq \frac{1}{4}$. So v' is a constant multiple of v, and it suffices to analyze v'.

We have $v' = (v'_1, \ldots, v'_n)$, where v'_1 is the component parallel to e_1. In the rest of this subsection we will establish the following bounds on v':

Theorem 11 *(i) The component of v' that is parallel to the target vector e_1 is $v'_1 \geq u_1(\frac{1}{2} - O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}})) > \frac{u_1}{4}$. (ii) The component of v' that is orthogonal to e_1, namely $v'_\perp = v' - v'_1 e_1 = (0, v'_2, \ldots, v'_n)$, satisfies $\|v'_\perp\| = O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}}) u_1$.*

Given Theorem 11, the error rate of the hypothesis $\mathrm{sgn}(v \cdot x)$ under $\mathcal{U}$ is

$$\Pr[\mathrm{sgn}(v' \cdot x) \neq \mathrm{sgn}(x_1)] = \frac{\arctan\left(\frac{\|v'_\perp\|}{v'_1}\right)}{\pi} \leq \frac{\arctan(O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}}))}{\pi} = O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}}).$$

By Fact 11 the sample average vector $\overline{v}$ has $\Pr_{x \in \mathcal{U}}[\mathrm{sgn}(\overline{v} \cdot x) \neq \mathrm{sgn}(v \cdot x)] \leq \epsilon$ with probability at least $1 - \delta$, and we obtain Theorem 3.

Now we prove Theorem 11. Note that if $\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+$ is the zero vector then the theorem clearly holds, so we henceforth assume that $\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+$ is not the zero vector.

Fix any unit vector $w \in S^{n-1}$. Suppose that p is such that the vector $\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+$ points in the direction of w, i.e. $w = \frac{\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+}{\|\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+\|}$; let $\tau > 0$ denote $\|\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+\|$, so $v' = u/2 + \tau w$. To establish Theorem 11, it suffices to show that the desired bounds hold for any function p which satisfies (15) and is such that: (a) the vector $\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+$ points in the direction of w, and (b) the magnitude of $\tau = \|\mathsf{opt}_- u^- - \mathsf{opt}_+ u^+\|$ is as large as possible. (Since $u/2$ contributes zero to $v'_\perp$, we have that $\|v_\perp\|$ scales with τ and thus condition (ii) only becomes harder to satisfy as τ increases. If $w_1 > 0$ then condition (i) holds for any $\tau > 0$, and if $w_1 < 0$ then the larger τ is the more difficult it is to satisfy condition (i).) We let $\tau_{\max}$ denote this maximum possible value of τ; if we can show that $|\tau_{\max}| = O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}})u_1$, then since $v'_1 = \frac{u_1}{2} + \tau w_1$ and $v'_\perp = \tau(0, w_2, w_3, \ldots, w_n)$, this gives Theorem 11.

We upper bound $\tau_{\max}$ by considering an even more relaxed scenario. Let w be any unit vector in S^{n-1}. Let A be any subset of S^{n-1} and let B be any subset of $S^{n-1} \setminus A$ such that $\mathsf{opt}_A + \mathsf{opt}_B = \mathsf{opt}$, where $\mathsf{opt}_A = \int_{x \in A} p(x)\mathcal{U}(x)$ and $\mathsf{opt}_B = \int_{x \in B} p(x)\mathcal{U}(x)$. Let $u^A = \frac{\int_{x \in A} x p(x)\mathcal{U}(x)}{\mathsf{opt}_A}$ and $u^B = \frac{\int_{x \in B} x p(x)\mathcal{U}(x)}{\mathsf{opt}_B}$. Let $p : S^{n-1} \to [0, 1]$ be any function such that (i) equation (15) holds, and (ii) the vector $\mathsf{opt}_A u^A - \mathsf{opt}_B u^B$ points in the direction of w. If we can upper bound the magnitude of $\mathsf{opt}_A u^A - \mathsf{opt}_B u^B$, then this gives an upper bound on $\tau_{\max}$. (This is a more relaxed scenario because we are not requiring that $A \subseteq S^-$ and $B \subseteq S^+$.) But now a simple convexity argument shows that $\|\mathsf{opt}_A u^A - \mathsf{opt}_B u^B\|$ is maximized by taking A to be $\{x \in S^{n-1} : x \cdot w \geq y\}$ where y is chosen so that $\int_{x \in A} \mathcal{U}(x) = \frac{\mathsf{opt}}{2}$; taking B to be $-A$; and taking $p(x)$ to be 1 on $x \in (A \cup B)$ and 0 on $x \notin (A \cup B)$ (note that this gives $\mathsf{opt}_A = \mathsf{opt}_B = \frac{\mathsf{opt}}{2}$). Let τ_{MAX} be the value of $\|\mathsf{opt}_A u^A - \mathsf{opt}_B u^B\|$ that results from taking $A, B, \mathsf{opt}_A, \mathsf{opt}_B$ and p as described in the previous sentence; we will show that $\tau_{MAX} = O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}})u_1$ and thus prove Theorem 11.

It is clear that $\mathsf{opt}_A u^A = -\mathsf{opt}_B u^B$, so it suffices to bound $\|\mathsf{opt}_A u^A\| = \frac{\tau_{MAX}}{2}$. Let $y \in [0, 1]$ be the value specified above, so

$$\frac{\mathsf{opt}}{2} = \Pr_{x \in \mathcal{U}}[x \cdot w \geq y] = \frac{A_{n-2}}{A_{n-1}} \cdot \int_y^1 (1 - z^2)^{(n-3)/2} dz. \tag{16}$$

We have

$$\mathsf{opt}_A u^A = \int_{x \in A} x p(x)\mathcal{U}(x) = \left(\frac{A_{n-2}}{A_{n-1}} \cdot \int_y^1 z(1 - z^2)^{(n-3)/2} dz \right) w,$$

so it remains to show that $\gamma = O(\mathsf{opt}\sqrt{\log \frac{1}{\mathsf{opt}}})$ where $\gamma > 0$ is such that

$$\frac{A_{n-2}}{A_{n-1}} \cdot \int_y^1 z(1 - z^2)^{(n-3)/2} dz = \gamma u_1 \tag{17}$$

where y satisfies (16). We do this in the following two claims.

Claim 13 *Let ℓ be such that $y = \frac{\ell}{\sqrt{n}}$. Then $e^{-\ell^2/2} = \Theta(\mathsf{opt})$.*

Proof: We have $\int_y^1 (1-z^2)^{(n-3)/2} dz = \frac{\mathsf{opt}}{2A_{n-2}/A_{n-1}} = \Theta(\frac{\mathsf{opt}}{\sqrt{n}})$. Write $y = \frac{\ell}{\sqrt{n}}$. Piecewise approximating the integral in increments of $1/\sqrt{n}$ we have

$$\int_y^1 (1-z^2)^{(n-3)/2} dz \approx \sum_{j=\ell}^{\sqrt{n}} e^{-j^2/2} \cdot \frac{1}{\sqrt{n}} = \Theta(e^{-\ell^2/2}) \cdot \frac{1}{\sqrt{n}}.$$

Since this equals $\Theta(\frac{\mathsf{opt}}{\sqrt{n}})$, we have that $e^{-\ell^2/2} = \Theta(\mathsf{opt})$, which gives the claim. (Note that we have $\ell = \Theta(\sqrt{\log \frac{1}{\mathsf{opt}}}) \gg 1$, which is compatible with approximatingS the integral with a sum as done above.) ■

Claim 14 *We have* $\gamma = \Theta(\mathsf{opt}\sqrt{\log(1/\mathsf{opt})})$.

Proof: From Claim 12 we have $u_1 = \Theta(\frac{1}{\sqrt{n}})$. Since $\frac{A_{n-2}}{A_{n-1}} = \Theta(\sqrt{n})$, by Equation (17) we have that $\gamma = \Theta(n \cdot \int_y^1 z(1-z^2)^{(n-3)/2} dz)$. Since $y = \ell/\sqrt{n}$ where $\ell = \Theta(\sqrt{\log(1/\mathsf{opt})})$ (and more precisely $e^{-\ell^2/2} = \Theta(\mathsf{opt})$) by Claim 13, again a piecewise approximation with pieces of length $1/\sqrt{n}$ gives us

$$\int_y^1 z(1-z^2)^{(n-3)/2} dz \approx \sum_{j=\ell}^{\sqrt{n}} \frac{j}{\sqrt{n}} \cdot e^{-j^2/2} \cdot \frac{1}{\sqrt{n}} < \frac{1}{n} \sum_{j=\ell}^{\infty} j e^{-j^2/2} = \Theta(\frac{\ell e^{-\ell^2/2}}{n})$$

and thus $\gamma = \Theta(\mathsf{opt}\sqrt{\log 1/\mathsf{opt}})$ as desired. ■

6 Learning halfspaces in the presence of malicious noise

We now consider the problem of PAC learning an unknown origin-centered halfspace, under the uniform distribution on S^{n-1}, in the demanding *malicious noise model* introduced by Valiant [37] and subsequently studied by Kearns and Li [20] and many others.

We first define the malicious noise model. Given a target function f and a distribution $\mathcal{D}$ over X, a *malicious example oracle with noise rate* η is an oracle $\mathsf{EX}_\eta(f, \mathcal{D})$ that behaves as follows. Each time it is called, with probability $1-\eta$ the oracle returns a noiseless example $(x, f(x))$ where x is drawn from $\mathcal{D}$, and with probability η it returns a pair (x, y) about which nothing can be assumed; in particular such a "malicious" example may be chosen by a computationally unbounded adversary which has complete knowledge of f, $\mathcal{D}$, and the state of the learning algorithm when the oracle is invoked. We say that an algorithm *learns to error* ϵ *in the presence of malicious noise at rate* η *under the uniform distribution* if it satisfies the following condition: given access to $\mathsf{EX}_\eta(f, \mathcal{U})$ with probability $1-\delta$ the algorithm outputs a hypothesis h such that $\Pr_{x \in \mathcal{U}}[h(x) \neq f(x)] \leq \epsilon$.

Few positive results are known for learning in the presence of malicious noise. Improving on [37, 20] Decatur [10] gave an algorithm to learn disjunctions under any distribution that tolerates a noise rate of $O(\frac{\epsilon}{n} \ln \frac{1}{\epsilon})$. More recently, Mansour and Parnas studied the problem of learning disjunctions under product distributions in an "oblivious" variant of the malicious noise model [30], giving an algorithm that can tolerate a noise rate of $O(\epsilon^{5/3}/n^{2/3})$. We note that the Perceptron algorithm can be shown to tolerate malicious noise at rate $O(\epsilon/\sqrt{n})$ when learning an origin-centered halfspace under the uniform distribution $\mathcal{U}$ on S^{n-1}.

It is not difficult to show that the simple `Average` algorithm can also tolerate malicious noise at rate $O(\epsilon/\sqrt{n})$:

Theorem 12 *For any $\epsilon > 0$, algorithm* `Average` *(with $m = O(\frac{n^2}{\epsilon^2} \cdot \log \frac{n}{\delta})$) learns the class of origin-centered halfspaces to error ϵ in the presence of malicious noise at rate $\eta = O(\frac{\epsilon}{\sqrt{n}})$ under the uniform distribution.*

Proof: If there were no noise the true average vector (average of all positive examples) would be $(u_1, 0, \ldots, 0)$ where by Claim 12 we have $u_1 = \Theta(1/\sqrt{n})$. By Chernoff bounds, we may assume that the true frequency η' of noisy examples in the sample is at most $2\eta = O(\epsilon/\sqrt{n})$. Let v denote the average of the noiseless vectors in the sample; Chernoff bounds are easily seen to imply that we have $v_1 = \Theta(1/\sqrt{n})$ and $|v_i| \leq \frac{\epsilon}{n}$ for each $i = 2, \ldots, n$. Let z denote the average location of the malicious examples in the sample; since even malicious example must lie on S^{n-1} (for otherwise we could trivially identify and discard them), it must be the case that $\|z\| \leq 1$. From this it is easy to see that the average $\overline{v}$ of the entire sample must satisfy $\overline{v}_1 = \Theta(1/\sqrt{n}) - \epsilon/\sqrt{n} = \Theta(1/\sqrt{n})$ and $\sqrt{\overline{v}_2^2 + \cdots + \overline{v}_n^2} = O(\epsilon/\sqrt{n})$. We thus have $\Pr_{x \in \mathcal{U}}[\text{sgn}(\overline{v} \cdot x) \neq \text{sgn}(x_1)] = \alpha(\overline{v}, e_1)/\pi = \arctan(\frac{O(\epsilon/\sqrt{n})}{\Theta(1/\sqrt{n})})/\pi \leq \epsilon$. ■

As the main result of this section, in Section 6.1 we show that by combining the `Average` algorithm with a simple preprocessing step to eliminate some noisy examples, we can handle a higher malicious noise rate of $O(\frac{\epsilon}{(n \log n)^{1/4}})$; this is Theorem 4. This algorithm, which we call `TestClose`, is the following:

1. Draw examples from $\mathsf{EX}_\eta(f, \mathcal{U})$ until $m = O(\frac{n^2}{\epsilon^2} \log \frac{n}{\delta})$ positively labeled examples have been received; let $S = \{x^1, \ldots, x^m\}$ denote this set of examples.

2. Let $\rho = \sqrt{\frac{C}{n} \log \frac{m}{\delta}}$, where C is a fixed constant (specified later in Section 6.1). If any pair of examples x^i, x^j with $i \neq j$ has $\|x^i - x^j\| < \sqrt{2 - \rho}$, remove both x^i and x^j from S. (We say that such a pair of examples is *too close*.) Repeat this until no two examples in S are too close to each other. Let S' denote this "reduced" set of examples.

3. Now run `Average` on S' to obtain a vector $\overline{v}$, and return the hypothesis $h(x) = \text{sgn}(\overline{v} \cdot x)$.

The idea behind this algorithm is simple. If there were no noise, then all examples received by the algorithm would be independent uniform random draws from the positive half of S^{n-1}, and it is not difficult to show that with high probability no two examples would be too close to each other. Roughly speaking, the adversary controlling the noise would like to cause $\overline{v}$ to point as far away from the true target vector as possible; in order to do this his best strategy (if we were simply running the `Average` algorithm on the original data set S without discarding any points) would be to have all noisy examples be located at some single particular point $x^\star \in S^{n-1}$. However, our "closeness" test rules out this adversary strategy, since it would certainly identify all these collocated points as being noisy and discard them. Thus intuitively, in order to fool our closeness test, the adversary is constrained to place his noisy examples relatively far apart on S^{n-1} so that they will not be identified and discarded. But this means that the noisy examples cannot have a very large effect on the average vector $\overline{v}$, since intuitively placing the noisy examples far apart on S^{n-1} causes their vector average to have small magnitude and thus to affect the overall average $\overline{v}$ by only a small amount. The actual analysis in the proof of Theorem 4 uses bounds from the theory of sphere packing in $\mathbb{R}^n$ to make these intuitive arguments precise.

6.1 Proof of Theorem 4

Let $S_{bad} \subseteq S$ denote the set of "bad" examples in S that were chosen by the adversary, and let S_{good} be $S \setminus S_{bad}$, the set of "good" noiseless examples. Let S'_{bad} (S'_{good}, respectively) denote $S_{bad} \cap S'$

($S_{good} \cap S'$, respectively), i.e. the set of bad (good, respectively) examples that survive the closeness test in Step 2.

Let us write v'_{good} to denote the vector average of all points in S'_{good} and v'_{bad} to denote the vector average of all points in S'_{bad}. If we let η' denote $\frac{|S'_{bad}|}{|S'|}$, then we have that the overall vector average $\overline{v}$ of all examples in S' is $(1-\eta')v'_{good} + \eta' v'_{bad}$.

We first show that our closeness test does not cause us to discard any good examples:

Lemma 15 *With probability at least $1 - \frac{\delta}{4}$ we have $S'_{good} = S_{good}$.*

Proof: Let x' be any fixed point on S^{n-1}. We will show that a uniform example drawn from $\mathcal{U}$ lies within distance $\sqrt{2-\rho}$ of x' with probability at most $\frac{\delta}{4m^2}$. Since there are at most m examples in S_{good}, this implies that for any individual example $x^i \in S$, the probability that x^i lies too close to any example in S_{good} is at most $\frac{\delta}{2m}$; taking a union bound gives the lemma.

Without loss of generality we may take $x' = (1, 0, \ldots, 0)$. It is easy to see that for any $y = (y_1, \ldots, y_n) \in S^{n-1}$, we have $\|y - x'\| = \sqrt{2 - 2y_1}$ and thus $\|y - x'\| < \sqrt{2-\rho}$ if and only if $y > \rho/2$. But by Fact 10, we have that if y is drawn from $\mathcal{U}$, then

$$\Pr_{y \in \mathcal{U}}[y > \rho/2] = \frac{A_{n-2}}{A_{n-1}} \cdot \int_{\rho/2}^{1} (1-z^2)^{(n-3)/2} dz. \tag{18}$$

It is easy to verify from the definition of ρ that for a suitable absolute constant C, the integrand $(1-z^2)^{(n-3)/2}$ is at most $(1-(\rho/2)^2)^{(n-3)/2} \leq \frac{\delta}{4m^3}$ over the interval $[\rho/2, 1]$, and thus (since $A_{n-2}/A_{n-1} = \Theta(\sqrt{n}) < m$) we have that (18) is at most $\frac{\delta}{4m^2}$ as required. ■

The true noise rate is η, and the previous lemma implies that with probability $1 - \frac{\delta}{4}$ we do not throw away any good examples from S. Using Chernoff bounds, it is easy to show that with overall probability at least $1 - \frac{\delta}{2}$ we have $\eta' < 2\eta$.

Let v_{good} denote $\frac{1}{|S_{good}|}\sum_{x \in S_{good}} x$, the average location of the vectors in S_{good}. We have that the expected value of v_{good} is $(u_1, 0, \ldots, 0)$ where $u_1 = \Theta(\frac{1}{\sqrt{n}})$ is as defined in Claim 12. For $m = O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$, as in the proof of Fact 11, Chernoff bounds imply that with probability at least $1 - \frac{\delta}{4}$ we have that $(v_{good})_1 = \Theta(\frac{1}{\sqrt{n}})$ while $(v_{good})_i = O(\frac{\epsilon}{n})$ for each $i = 2, \ldots, n$. By Lemma 15, with probability at least $1 - \frac{\delta}{4}$ we have $v'_{good} = v_{good}$, so with overall probability at least $1 - \frac{\delta}{2}$ we have $(v'_{good})_1 = \Theta(\frac{1}{\sqrt{n}})$ while $(v'_{good})_i = O(\frac{\epsilon}{n})$ for each $i = 2, \ldots, n$.

We now show that $\|v'_{bad}\|$ must be small; once we establish this, as we will see it is straightforward to combine this with the bounds of the previous two paragraphs to prove Theorem 4. The desired bound on $\|v'_{bad}\|$ is a consequence of the following lemma:

Lemma 16 *Let T be any set of $M = \omega(n^{3/2}/\sqrt{\rho})$ many examples on S^{n-1} such that no two examples in T lie within distance $\sqrt{2-\rho}$ of each other (recall that $\rho = \sqrt{\frac{C}{n}\log\frac{m}{\delta}}$). Then the vector average $t = \frac{1}{|T|}\sum_{x \in T} x$ of T satisfies $\|t\| = O\left(\frac{(\log\frac{m}{\delta})^{1/2}}{n^{1/4}}\right)$.*

Proof: Without loss of generality we may suppose that $t = (c, 0, \ldots, 0)$ for some $c > 0$ (by rotating the set T); our goal is to upper bound c. We consider a partition of T based on the value of the first coordinate as follows. For $\tau = 1, 1 - \frac{1}{\sqrt{n}}, 1 - \frac{2}{\sqrt{n}}, \ldots$ we define the set T_τ to be $\{x \in T : \tau - \frac{1}{2\sqrt{n}} \leq x_i < \tau + \frac{1}{2\sqrt{n}}\}$. The idea of the proof is that for any value of τ which is not very small, the set T_τ must be small because of sphere-packing bounds. This implies that the overwhelming majority of the M examples in T must have a small first coordinate, which gives the desired result.

More precisely, we have the following claim:

Claim 17 *There is a fixed constant $K > 0$ such that if $\tau > K\sqrt{\rho}$, then $|T_\tau| \le n$.*

Proof: We first give a crude argument to show that that if $\tau > 0.1$ then $|T_\tau| \le n$. (It will be clear from the argument that any positive constant could be used in this argument instead of 0.1.) This argument uses the same basic ideas as the general case of $\tau > K\sqrt{\rho}$ but is simpler because we do not need our bounds to be as precise; later for the general case it will be useful to be able to assume that $\tau < 0.1$.

Fix some $\tau > 0.1$. We first note that if τ is greater than (say) $4/5$ then T_τ can contain at most one point (since any two points of S^{n-1} which both have first coordinate $4/5 \pm o(1)$ can have Euclidean distance at most $6/5 + o(1) < \sqrt{2-\rho}$ from each other). Thus we may assume that $0.1 < \tau < 4/5$ (the key aspect of the upper bound is that τ is bounded away from 1).

For $x \in \mathbb{R}^n$ let x' denote $(x_2, \ldots, x_n)$. Since each $x \in T_\tau$ has $x_1 \in [\tau - \frac{1}{2\sqrt{n}}, \tau + \frac{1}{2\sqrt{n}})$, we have that each $x \in T_\tau$ satisfies $\|x'\| = \sqrt{1-\tau^2} \cdot (1 \pm o(1))$. Let $\tilde{x}' \in \mathbb{R}^{n-1}$ denote the rescaled version of x' so that $\|\tilde{x}'\|$ equals $\sqrt{1-\tau^2}$ exactly, and let $\tilde{T}'_\tau$ denote $\{\tilde{x}' : x \in T_\tau\}$. Since the first coordinates of any two points in T_τ differ by at most $\frac{1}{\sqrt{n}}$, it is not difficult to see that that the minimum pairwise distance condition on T_τ implies that any pair of points in $\tilde{T}'_\tau$ must have distance at least $(\sqrt{2-\rho} - \frac{1}{\sqrt{n}}) \cdot (1 - o(1)) = \sqrt{2} \cdot (1 - o(1))$ from each other.

We now recall *Rankin's second bound* on the minimum pairwise distance for point sets on Euclidean spheres (see e.g. Theorem 1.4.2 of [11]). This bound states that for any value $\kappa > \sqrt{2}$, at most $n + 1$ points can be placed on S^{n-1} if each point is to have distance at least κ from all other points. By rescaling, this immediately implies that at most n points can be placed on the Euclidean sphere of radius $\sqrt{1-\tau^2}$ in $\mathbb{R}^{n-1}$ if all pairwise distances are at least $\kappa\sqrt{1-\tau^2}$. Now recall from the previous paragraph that all points in $\tilde{T}'_\tau$ lie on the sphere of radius $\sqrt{1-\tau^2}$, and all pairwise distances in in $\tilde{T}'_\tau$ are at least $\sqrt{2} \cdot (1 - o(1))$. It follows by a suitable choice of $\kappa > \sqrt{2}$ that $|\tilde{T}'_\tau|$, and thus $|T_\tau|$, is at most n.

We henceforth assume that $K\sqrt{\rho} < \tau < 0.1$, and give a more quantitatively precise version of the above argument to handle this case. We consider the following transformation f that maps points in T_τ onto the ball of radius $\sqrt{1-\tau^2}$ in $\mathbb{R}^{n-1}$: given $x = (x_1, \ldots, x_n) \in T_\tau$, let

$$f(x) = \sqrt{1-\tau^2} \cdot \frac{x'}{\|x'\|}$$

i.e. $f(x)$ is obtained by removing the first coordinate and normalizing the resulting $(n-1)$-dimensional vector to have magnitude $\sqrt{1-\tau^2}$.

We now claim that if $x \neq y$, $x, y \in T_\tau$, then we have $\|f(x) - f(y)\| > \sqrt{2-\rho} - \frac{1}{\sqrt{n}} - \frac{3\tau^2}{5}$. To see this, fix any $x, y \in T_\tau$. By the triangle inequality we have

$$\|f(x) - f(y)\| \ge \|x' - y'\| - \|f(x) - x'\| - \|f(y) - y'\|, \tag{19}$$

so it suffices to bound the terms on the right hand side.

For the first term, we have

$$\sqrt{2-\rho} \le \|x - y\| \le \frac{1}{\sqrt{n}} + \sqrt{(x_2 - y_2)^2 + \cdots + (x_n - y_n)^2},$$

where the first inequality holds since $x, y \in T$ and the second inequality holds since the first coordinates of x and y differ by at most $\frac{1}{\sqrt{n}}$. This immediately gives $\|x' - y'\| \geq \sqrt{2-\rho} - \frac{1}{\sqrt{n}}$.

For the second term, since $x_1 \in [\tau - \frac{1}{2\sqrt{n}}, \tau + \frac{1}{2\sqrt{n}})$, it must be the case that

$$\|x'\|^2 = x_2^2 + \cdots + x_n^2 \in \left(1 - (\tau + \frac{1}{2\sqrt{n}})^2, 1 - (\tau - \frac{1}{2\sqrt{n}})^2\right]. \tag{20}$$

We have

$$\|f(x) - x'\| = \left\| \frac{(1-\tau^2)^{1/2}}{\|x'\|} x' - x' \right\| = \left| \frac{(1-\tau^2)^{1/2}}{\|x'\|} - 1 \right| \cdot \|x'\| \leq \left| \frac{(1-\tau^2)^{1/2}}{\|x'\|} - 1 \right| \tag{21}$$

where the last inequality uses $\|x'\| \leq 1$. A tedious but straightforward verification (using the fact that $\tau < 0.1$) shows that condition (20) implies that the right side of (21) is at most $\frac{\tau^2}{10}$ (see Section 6.1.1 for the proof). The third term $\|f(y) - y'\|$ clearly satisfies the same bound.

Combining the bounds we have obtained, it follows from (19) that $\|f(x) - f(y)\| \geq \sqrt{2-\rho} - \frac{1}{\sqrt{n}} - \frac{\tau^2}{5}$. For some fixed absolute constant $K > 0$, we have that if $\tau^2 > K^2\rho$ (i.e. $\tau > K\sqrt{\rho}$), then the right side of this last inequality is at least $\sqrt{2} - \frac{\tau^2}{2}$. So we have established that the transformed set of points $f(T_\tau)$ have all pairwise distances at least $\sqrt{2} - \frac{\tau^2}{2}$. But just as in the crude argument at the beginning of the proof, Rankin's bound implies that any point set on the radius-$\sqrt{1-\tau^2}$ ball in $\mathbb{R}^{n-1}$ with all pairwise distances strictly greater than $\sqrt{2} \cdot \sqrt{1-\tau^2}$ must contain at most n points. Since (as is easily verified) $\sqrt{2} - \frac{\tau^2}{2} > \sqrt{2} \cdot \sqrt{1-\tau^2}$, it must be the case that $|T_\tau| \leq n$. (Claim 17)■

With Claim 17 in hand, it is clear that at most $n^{3/2}$ examples $x \in T$ can have $x_1 \geq K\sqrt{\rho}$. Since certainly each point in T has first coordinate at most 1, the average value of the first coordinate of all M points in T must be at most

$$\frac{n^{3/2} + MK\sqrt{\rho}}{M} \leq 2K\sqrt{\rho} = \Theta\left(\frac{(\log \frac{m}{\delta})^{1/4}}{n^{1/4}}\right)$$

(where we used $M = \omega(n^{3/2}/\sqrt{\rho})$ for the inequality above), and Lemma 16 is proved. ■

Lemma 16 implies that $\|v'_{bad}\| = O(\frac{(\log \frac{m}{\delta})^{1/4}}{n^{1/4}})$. (Note that if S'_{bad} is not of size M, we can augment it with examples from S'_{good} in order to make it large enough so that we can apply the lemma. This can easily be done since we only need $M = \tilde{\omega}(n^{7/4})$ for the lemma and we have $|S_{good}| = \tilde{\Theta}(\frac{n^2}{\epsilon^2})$.) Putting all the pieces together, we have that with probability $1-\delta$ all the following are true:

- $(v'_{good})_1 = \Theta(\frac{1}{\sqrt{n}})$;
- $(v'_{good})_i = O(\frac{\epsilon}{n})$ for $i = 2, \ldots, n$;
- $\|v'_{bad}\| = O(\frac{(\log \frac{m}{\delta})^{1/4}}{n^{1/4}})$;
- $\eta' \leq 2\eta$, where $\overline{v} = (1-\eta')v'_{good} + \eta' v'_{bad}$.

Combining all these bounds, a routine analysis shows that the angle between $\overline{v}$ and the target $(1, 0, \ldots, 0)$ is at most ϵ provided that

$$\frac{(2\eta)\frac{\log^{1/4}(m/\delta)}{n^{1/4}}}{\frac{1}{\sqrt{n}}} \leq c \cdot \epsilon$$

for some sufficiently small constant c. Rearranging this inequality, Theorem 4 is proved.

6.1.1 Proof that (21) is at most $\frac{\tau^2}{10}$

We have that (21) $\leq \left|\frac{(1-\tau^2)^{1/2}}{\|x'\|} - 1\right|$. To bound this quantity we will consider the largest value greater than 1 and smallest value less than 1 that $\frac{(1-\tau^2)^{1/2}}{\|x'\|}$ can take. Throughout the following bounds we repeatedly use the fact that $0 < \tau < 0.1$.

We have that

$$\|x'\| > \sqrt{1 - (\tau + \frac{1}{2\sqrt{n}})^2} > 1 - \frac{9\tau^2}{16}$$

where the first inequality is from (20) and the second is easily verified (recall that $\tau > K\sqrt{\rho} > \frac{1}{n^{1/4}}$). Since $(1-\tau^2)^{1/2} < 1 - \frac{\tau^2}{2}$, we have $\frac{(1-\tau^2)^{1/2}}{\|x'\|} < \frac{1-\tau^2/2}{1-9\tau^2/16} = 1 + \frac{\tau^2/16}{1-9\tau^2/16} < 1 + \frac{\tau^2}{10}$.

On the other hand, from (20) we also have that $\|x'\| \leq \sqrt{1-(\tau - \frac{1}{2\sqrt{n}})^2}$, so consequently we have (writing b for $\frac{1}{2\sqrt{n}}$ for readability below):

$$\frac{(1-\tau^2)^{1/2}}{\|x'\|} \geq \sqrt{\frac{1-\tau^2}{1-(\tau-b)^2}} = \sqrt{1 - \frac{2b\tau - b^2}{1-(\tau-b)^2}} > 1 - \frac{3}{5} \cdot \frac{2b\tau - b^2}{1-(\tau-b)^2}.$$

Recalling that $b = \frac{1}{2\sqrt{n}}$ whereas $\frac{1}{n^{1/4}} < \tau < 0.1$, we see that $\frac{3}{5} \cdot \frac{2b\tau-b^2}{1-(\tau-b)^2}$ is greater than 0 but is easily less than $\frac{\tau^2}{10}$.

We thus have that $\left|\frac{(1-\tau^2)^{1/4}}{\|x'\|} - 1\right| < \frac{\tau^2}{10}$ as claimed.

7 Directions for Future Work

There are many natural ways to extend our work. One promising direction is to try to develop a broader range of learning results over the sphere S^{n-1} using the Hermite polynomials basis, in analogy with the rich theory of uniform distribution learning that has been developed for the parity basis over $\{-1,1\}^n$. Another natural goal is to gain a better understanding of the distributions and concept classes for which we can use the polynomial regression algorithm as an agnostic learner. Is there a way to extend the analysis of the $d = 1$ case of the polynomial regression algorithm (establishing Theorem 3) to obtain a stronger version of Theorem 1, Part 1(b)? Another natural idea would be to use the "kernel trick" with the polynomial kernel to speed up the algorithm. Finally, we intend to explore whether the polynomial regression algorithm can be used for other challenging noisy learning problems beyond agnostic learning, such as learning with malicious noise.

References

[1] E. Baum. The Perceptron algorithm is fast for nonmalicious distributions. *Neural Computation*, 2:248–260, 1990.

[2] E. B. Baum and Y-D. Lyuu. The transition to perfect generalization in perceptrons. *Neural Computation*, 3:386–401, 1991.

[3] A. BLum. Machine learning: a tour through some favorite results, directions, and open problems. FOCS 2003 tutorial slides, available at http://www-2.cs.cmu.edu/~avrim/Talks/FOCS03/tutorial.ppt, 2003.

[4] 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.

[5] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[6] Avrim Blum, Merrick L. Furst, Jeffrey Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proc. 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–262. ACM Press, 1994.

[7] S. Bonan and D. Clark. Estimates of the hermite and the freud polynomials. *Journal of Approximation Theory*, 63:210–224, 1990.

[8] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

[9] Kenneth L. Clarkson. Subgradient and sampling algorithms for l1 regression. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 257–266, 2005.

[10] S. Decatur. Statistical queries and faulty PAC oracles. In *Proceedings of the Sixth Workshop on Computational Learning Theory*, pages 262–268, 1993.

[11] T. Ericson and V. Zinoviev. *Codes on Euclidean Spheres*. North-Holland Mathematical Library, 2001.

[12] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[13] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of the Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[14] S. Goldman, M. Kearns, and R. Schapire. On the Sample Complexity of Weakly Learning. *Information and Computation*, 117(2):276–287, 1995.

[15] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[16] J. Jackson. *The Harmonic sieve: a novel application of Fourier analysis to machine learning theory and practice*. PhD thesis, Carnegie Mellon University, August 1995.

[17] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[18] J. Jackson, A. Klivans, and R. Servedio. Learnability beyond AC^0. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 776–784, 2002.

[19] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

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

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

[22] A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004. Preliminary version in *Proc. of FOCS'02*.

[23] A. Klivans and R. Servedio. Boosting and hard-core sets. In *Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science*, pages 624–633, 1999.

[24] W. Lee, P. Bartlett, and R. Williamson. Efficient agnostic learning of neural networks with bounded fan-in. *IEEE Transactions on Information Theory*, 42(6):2118–2132, 1996.

[25] Wee Sun Lee, Peter L. Bartlett, and Robert C. Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 369–376, Santa Cruz, California, 5–8 July 1995. ACM Press.

[26] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[27] P. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995.

[28] P. Long. An upper bound on the sample complexity of pac learning halfspaces with respect to the uniform distribution. *Information Processing Letters*, 87(5):229–234, 2003.

[29] L. Lovász and S. Vempala. Logconcave functions: Geometry and efficient sampling algorithms. In *Prooceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 650–659, 2003.

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

[31] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. In *Proceedings of the Twenty-Fourth Annual Symposium on Theory of Computing*, pages 462–467, 1992.

[32] R. O'Donnell and R. Servedio. New degree bounds for polynomial threshold functions. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 325–334, 2003.

[33] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proceedings of the 24th Symposium on Theory of Computing*, pages 468–474, 1992.

[34] R. Servedio. On PAC learning using Winnow, Perceptron, and a Perceptron-like algorithm. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 296–307, 1999.

[35] Gabor Szegö. *Orthogonal Polynomials*, volume XXIII of *Americam Mathematical Society Colloquium Publications*. A.M.S, Providence, 1989.

[36] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

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

[38] V. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, New York, 1998.

A Solving L_1 polynomial regression in polynomial time

Let $\mathcal{S}$ denote the set of all indices of monomials of degree at most d over variables $x_1, \ldots, x_n$, so $|\mathcal{S}| \leq n^{d+1}$. Our goal is to find $w_S \in \mathbb{R}$ for $S \in \mathcal{S}$ to minimize $\frac{1}{m} \sum_{i=1}^{m} |y^i - \sum_{S \in \mathcal{S}} w_S (x^i)_S|$, where x_S is the monomial indexed by S. This can be done by solving the following LP:

$$\min \sum_{i=1}^{m} z_i \quad \text{such that} \quad \forall i : \qquad \begin{aligned} & z_i \geq y^i - \sum_{S \in \mathcal{S}} w_S (x^i)_S \qquad \text{and} \\ & z_i \geq -\left(y^i - \sum_{S \in \mathcal{S}} w_S (x^i)_S \right). \end{aligned}$$

Using polynomial-time algorithm for linear programming this can be solved exactly in $n^{O(d)}$ time. In fact, for our purposes it is sufficient to obtain an approximate minimum, and hence one can use even more efficient algorithms [9].