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

Abstract

We give the first algorithm that (under distributional assumptions) efficiently learns halfspaces in the notoriously difficult agnostic *framework of Kearns, Schapire, & Sellie, where a learner is given access to labeled examples drawn from a distribution, without restriction on the labels (e.g. adversarial noise). The algorithm 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 $\epsilon > 0$, under the uniform distribution over $\{-1,1\}^n$ or the unit sphere in $\mathbb{R}^n$, as well as under any log-concave distribution over $\mathbb{R}^n$. It also agnostically learns Boolean disjunctions in time $2^{\tilde{O}(\sqrt{n})}$ with respect to* any *distribution. The new algorithm, essentially L_1* polynomial regression, *is a noise-tolerant arbitrary-distribution generalization of the "low-degree" Fourier algorithm of Linial, Mansour, & Nisan. We also give a new algorithm for PAC learning halfspaces under the uniform distribution on the unit sphere with the current best bounds on tolerable rate of "malicious noise."*

1. Introduction

Halfspaces have been used extensively in Machine Learning for decades. From the early work on the Perceptron algorithm in the 1950's, through the learning of artificial neural networks in the 1980's, and up to and including today's Adaboost [9] and Support Vector Machines [31], halfspaces have played a central role in the development of the field's most important tools.

Formally, a *halfspace* is a Boolean function $f(x) = \mathrm{sgn}(\sum_{i=1}^{n} w_i x_i - \theta)$. 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?

1.1. Agnostic Learning

The *agnostic learning* framework, introduced by Kearns et al. [16], 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 [29]) 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

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

†This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, by a grant no. 1079/04 from the Israel Science Foundation and an IBM faculty award. This publication only reflects the authors' views.

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

labels given to the learning algorithm. The goal is to find a hypothesis h with error $\Pr_{\mathcal{D}}[h(x) \neq y]$ as close as possible to η, efficiently in the dimension n (such problems in $\mathbb{R}^n$ can often be done in time $\exp(n)$). 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 [4, 14].

Unfortunately, only few positive results are known for agnostically learning expressive concept classes. Kearns *et al.* [16] gave an algorithm for agnostically learning piecewise linear functions, and Goldman *et al.* [10] showed how to agnostically learn certain classes of geometric patterns. Lee *et al.* [18] 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 [16]. In fact, it is known [19] 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 *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 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 $\{-1, 1\}^n$ [20, 6, 11, 12, 17]. For learning with respect to the uniform distribution on the unit sphere, our analysis uses the Hermite polynomials [28], 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}.

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 $\mathbb{R}^n \times \{-1, 1\}$ and let* opt *be the error rate of the best halfspace. 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.* [17]. 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 previously. (We note that learning halfspaces under the uniform distribution on S^{n-1} is a well-studied problem, see e.g. [1, 2, 14, 21, 22].) As before, we show that a related algorithm gives a hypothesis with error $O(\mathsf{opt} + \epsilon)$ in time $n^{O(1/\epsilon^2)}$.

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, or 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:

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

In Section 5.1 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 5.2 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 5.2). 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 5.2 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 $O(\frac{\epsilon}{n^{1/4}\log^{1/2}(n/\epsilon)})$. 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.

Finally, in Section 5.3 we show that the polynomial regression algorithm can be applied in non-noisy settings. We obtain a slightly better running time bound than the algorithm of Klivans *et al.* [17] for learning an intersection of halfspaces under the uniform distribution on $\{-1,1\}^n$.

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 [16] 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 $\mathbb{R}$.

We assume that our algorithms are given m examples $(x^1, y^1), \ldots, (x^m, y^m)$ 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 $z < 0$. Lastly, we define the set $\mathbb{P}_d$ to be the set of univariate polynomials of degree at most d.

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.* [20], 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 $|S| \leq d$ from the data, and constructing the degree-d polynomial $p(x) = \sum_{|S|\leq d} \tilde{f}(S)\chi_S(x)$ 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 $\Pr_\mathcal{D}[\mathrm{sgn}(p(x)) \neq f(x)] \leq \mathbf{E}_\mathcal{D}[(p(x) - f(x))^2]$ 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 $n \geq 1$, all $0 < \epsilon < \frac{1}{2}$, and all $f \in \mathcal{C}_n$ we have $\sum_{|S|\geq\alpha(\epsilon,n)} \hat{f}(S)^2 \leq \epsilon$. The low-degree algorithm is useful because it efficiently constructs a high-accuracy approximator for functions that have good Fourier concentration bounds:

Fact 1 ([20]) *Let $\mathcal{C}$ be a concept-class with concentration bound $\alpha(\epsilon, n)$. 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 $\mathbf{E}[(p(x) - f(x))^2] \leq \epsilon$ and runs in time* $\mathrm{poly}(n^{\alpha(\epsilon/2,n)}, \log\frac{1}{\delta})$.

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 $\sum_{|S|\geq\alpha(\epsilon/2,n)} = \mathbf{E}[(p(x) - f(x))^2] \leq \epsilon/2$. 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 [20] 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 $(x, y) \in X \times \{-1, 1\}$. In least-squares ($L_2$) polynomial regression, one attempt to minimize the following:

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

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 $\leq d$ monomials, $p(x) = \sum_b p_b \prod_{i=1}^n (x_i)^{b_i}$ where the sum is over $\{b \in \mathbb{Z}^n | \sum_{i=1}^n b_i \leq d, \forall i\ b_i \geq 0\}$. 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 $\leq n^{d+1}$ 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 the previous displayed equation 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 $S \subseteq [n], |S| \leq d$. Using Parseval's identity, it is not difficult to show that best degree d polynomial is exactly

$$\arg\min_{p \in \mathbb{P}_d} \mathbf{E}_{\mathcal{U}_X}\Big[(p(x) - f(x))^2\Big] = \sum_{S \subseteq [n]:|S| \leq d} \hat{f}(S)\chi_S(x),$$

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.* [16] prove the following statement about agnostic learning with the low-degree algorithm:

Fact 2 ([16], 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 [16] because as long as opt is bounded away from 1/2, this resulting hypothesis has error bounded from 1/2. However, the above bound is $1/4$ for $\mathsf{opt} = 0$. We now show 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 $n^{O(\alpha(\epsilon/3, n))}$.*

Proof sketch: Let $f \in C$ be an optimal function, i.e. $\Pr[y \neq f(x)] = \mathsf{opt}$. As described above, the low-degree algorithm (approximately) finds the best degree-d approximation $p(x)$ to the data y, i.e. $\min_{p \in \mathbb{P}_d} \mathbf{E}[(p(x) - y)^2]$, and the same term represents the mean squared error of p. This can be bounded using an "almost-triangle" inequality for $a, b, c \in \mathbb{R}$, that $\frac{1}{2}(a - c)^2 \leq (a - b)^2 + (b - c)^2$. Setting $a = \sum_{|S|<d} \hat{f}(S)\chi_S(x)$, $b = f(x)$, and $c = y$, we have that $\frac{1}{2}\min_{p \in \mathbb{P}_d} \mathbf{E}[(p(x) - y)^2]$ is at most,

$$\mathbf{E}\left[(f(x) - y)^2 + \left(\textstyle\sum_{|S|<d}\hat{f}(S)\chi_S(x) - f(x)\right)^2\right].$$

The above can be rewritten as,

$$\frac{1}{2}\min_{p \in \mathbb{P}_d} \mathbf{E}[(p(x) - y)^2] \leq 4\Pr[y \neq f(x)] + \sum_{|S| \geq d} \hat{f}(S)^2.$$

The first term is $4\mathsf{opt}$ and the second is at most $\epsilon/3$ for $d = \alpha(n, \epsilon/3)$. Outputting $h(x) = \mathrm{sgn}(p(x))$ would give error,

$$\Pr[\mathrm{sgn}(p(x)) \neq y] \leq \mathbf{E}[(p(x) - y)^2] \leq 8\mathsf{opt} + \frac{2}{3}\epsilon.$$

This leaves an additional $\epsilon/3$ for sampling error. ■

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

3. L_1 polynomial regression

Given the setup in Sections 2.2 and 2.3, 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 L_2 regression, in L_1 polynomial regression we attempt to minimize:

$$\min_{p \in \mathbb{P}_d} \mathbf{E}_{\mathcal{D}}\left[|p(x) - y|\right] \approx \min_{p \in \mathbb{P}_d} \frac{1}{m}\sum_{j=1}^m \left|p(x^j) - y^j\right|. \quad (3)$$

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 [7], and the minimizing polynomial p for the RHS of (3) can be obtained in poly(n^d) time using linear programming. 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 $\mathsf{opt} + \epsilon$.

L_1 **polynomial regression**(d, m, r)**:**

1. Take examples $(x^1, y^1), \ldots, (x^m, y^m)$.
2. Find polynomial p of degree $\le d$ 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 $\le d$ and then performing L_1 *linear* regression, as described earlier.)
3. Let $h(x) = \mathrm{sgn}(p(x) - t)$ for threshold $t \in [-1, 1]$ chosen uniformly at random.
4. Repeat the above three steps r times, each time with m fresh examples, and output the hypothesis with lowest error on its own data set.

Theorem 5 *Say* $\min_{p \in \mathbb{P}_d} E_{\mathcal{D}_X}[(p(x) - c(x))^2] \le \epsilon^2$ *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, with probability* $1 - \delta$*, using* $r = 4\log(2/\delta)/\epsilon$ *repetitions of* $m = \mathrm{poly}(n^d/\epsilon, \log 1/\delta)$ *examples each, the* L_1 *polynomial regression algorithm outputs a hypothesis* $h(x)$ *such that,* $\Pr_{\mathcal{D}}[h(x) \ne y] \le \mathsf{opt} + \epsilon$.

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: Let c be an optimal classifier in $\mathcal{C}$. Using the triangle inequality and $E[|Z|] \le \sqrt{E[Z^2]}$ for any random variable Z, we see that the quantity $\min_{p \in \mathbb{P}_d} \mathbf{E}_{\mathcal{D}}[|y - p(x)|]$ is at most:

$$\mathbf{E}_{\mathcal{D}}[|y - c(x)|] + \min_{p \in \mathbb{P}_d} \mathbf{E}_{\mathcal{D}}[|c(x) - p(x)|] \le 2\mathsf{opt} + \epsilon.$$

This is also an upper bound on the expected empirical error on any single iteration of steps 1, 2, and 3, i.e.,

$$\mathbf{E}\left[\frac{1}{m}\sum_{i=1}^{m} \left|y^i - p(x^i)\right|\right] \le 2\mathsf{opt} + \epsilon.$$

Next observe that, for any $y \in \{-1, 1\}, z \in \mathbb{R}$,

$$\Pr_{t \in [-1,1]}[y \ne \mathrm{sgn}(z - t)] = \begin{cases} \frac{1}{2}|y - z| & |z| \le 1 \\ \frac{1}{2}|y - \mathrm{sgn}(z)| & |z| > 1 \end{cases}$$

In either case, the right hand side is at most $\frac{1}{2}|y - z|$. Thus, on any single iteration,

$$\mathbf{E}\left[\frac{|\{i \mid y^i \ne h(x^i)\}|}{m}\right] \le \frac{2\mathsf{opt} + \epsilon}{2} = \mathsf{opt} + \frac{\epsilon}{2}.$$

By Markov's inequality, on any single iteration,

$$\Pr\left[\frac{|\{i \mid y^i \ne h(x^i)\}|}{m} \ge \mathsf{opt} + \frac{2\epsilon}{3}\right] \le \frac{\mathsf{opt} + \frac{\epsilon}{2}}{\mathsf{opt} + \frac{2\epsilon}{3}}.$$

WLOG $\mathsf{opt} + \epsilon \le 1/2$ (otherwise the lemma is trivial), in which case the above is at most $(1 + \epsilon/3)^{-1}$. Hence, after $4\log(2/\delta)/\epsilon$ repetitions, with probability at most $(1 + \epsilon/3)^{-4\log(2/\delta)/\epsilon} \le \delta/2$ (using $(1 + \epsilon/3)^{4/\epsilon} \le 1/e$ for $\epsilon \in [0, 1]$), one of the repetitions will have empirical error at most $|\{i \mid y^i \ne h(x^i)\}|/m \le \mathsf{opt} + (2/3)\epsilon$.

Now, suppose this is the case. Next, note that our output hypothesis is a halfspace over n^d attributes. By VC theory, for $m = \mathrm{poly}(n^d/\epsilon, \log(2r/\delta))$, with probability $1 - \delta/(2r)$, no such halfspace will have generalization error more than $\epsilon/3$ larger than its training error. Taking the union bound over all r repetitions, with probability $1 - \delta$, we have error at most $\mathsf{opt} + \epsilon$. ■

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.

4. Agnostic learning halfspaces and disjunctions via polynomial regression

In this section we sketch 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 [17]:

Fact 5 *[17] 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 $p_d(x) \approx \mathrm{sgn}(x)$, and say we have an n-dimensional halfspace $\mathrm{sgn}(v \cdot x - \theta)$. Then, $\mathrm{sgn}(v \cdot x - \theta) \approx p_d(v \cdot x - \theta)$. 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*

$$\int_{-\infty}^{\infty} (p_{d,\theta}(x) - \mathrm{sgn}(x-\theta))^2 \frac{e^{-x^2}}{\sqrt{\pi}} dx = O\left(\frac{1}{\sqrt{d}}\right). \quad (4)$$

The complete proof of this theorem, as well as the analysis of log-concave distributions, is available in the full version of the paper [13].

Proof sketch: WLOG $\theta \in [0, \sqrt{d}]$. For $\theta > \sqrt{d}$, it can be shown that the constant polynomial $p(x) = -1$ will be a sufficiently good approximation of $\mathrm{sgn}(x-\theta)$. For $\theta < 0$, an entirely similar proof holds.

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 (4), 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 (4) are,

$$\begin{aligned} c_i &= \int_{-\infty}^{\infty} \mathrm{sgn}(x-\theta) \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \\ &= \int_{\theta}^{\infty} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx - \int_{-\infty}^{\theta} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \\ &= 2 \int_{\theta}^{\infty} \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx \quad \text{(for } i \geq 1\text{)} \qquad (5) \end{aligned}$$

The last step follows from the fact that $\int_{-\infty}^{\infty} \mathrm{sgn}(x - \theta) \bar{H}_i(x) \frac{e^{-x^2}}{\sqrt{\pi}} dx = 0$ for $i \geq 1$ by orthogonality of $\bar{H}_i$ with $\bar{H}_0$. Next, the LHS of (4) is exactly $\sum_{i=d+1}^{\infty} c_i^2$.

It is straightforward to calculate each coefficient c_i using standard properties of the Hermite Polynomials. It is well known [28] that the Hermite polynomials can be defined by: $H_i(x) e^{-x^2} = (-1)^i \frac{d^n}{dx^n} e^{-x^2}$, which implies $\frac{d}{dx} H_i(x) e^{-x^2} = -H_{i+1}(x) e^{-x^2}$. In turn, this and (5) imply that for $i \geq 1$,

$$\begin{aligned} c_i &= \frac{2}{\sqrt{\pi 2^i i!}} \int_{\theta}^{\infty} H_i(x) e^{-x^2} dx \\ &= \frac{2}{\sqrt{\pi 2^i i!}} \left(-H_{i-1}(x) e^{-x^2} \right) \Big|_{\theta}^{\infty} \\ &= \frac{2}{\sqrt{\pi 2^i i!}} H_{i-1}(\theta) e^{-\theta^2}. \qquad (6) \end{aligned}$$

To show that $\sum_{i=d+1}^{\infty} c_i^2 = O(1/\sqrt{d})$, it suffices that $c_i^2 = O(i^{-3/2})$. This follows from (6) and Theorem 1.i of [5], for $\theta < \sqrt{d}$. ∎

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 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.

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 *[26, 24, 17] 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 $|f(x) - p(x)| \le \epsilon$.*

For $\epsilon = \Theta(1)$ this fact appears in [26, 24]; an easy extension to arbitrary ϵ is given in [17]. 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 some existence results are known for low-degree L_∞-approximators of richer concept classes than just disjunctions. For example, results of O'Donnell and Servedio [25] show that any Boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ 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.

5. Extensions and Other Applications

5.1. 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 $d = 1$ version of the L_2 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 $i \ne j$ 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 [27] 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. The proof is available in the full version of the paper [13].

5.2. 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 [30] and subsequently studied by Kearns and Li [15] 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 $\Pr_{x \in \mathcal{U}}[h(x) \ne f(x)] \le \epsilon$.

Only few positive results are known for learning in the presence of malicious noise. Improving on [30, 15] Decatur [8] 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 [23], 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 $O(\epsilon/\sqrt{n})$ (see the full version [13] for the proof). We now 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 $\Omega(\frac{\epsilon}{(n\log n)^{1/4}})$. 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. If any pair of examples x^i, x^j with $i \neq j$ has $\|x^i - x^j\| < \sqrt{2-\rho}$, remove 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 $h(x) = \mathrm{sgn}(\overline{v}\cdot x)$.

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 S^{n-1}, and it is not difficult to show that with very 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, in the full version [13], uses bounds from the theory of sphere packing in $\mathbb{R}^n$ to make these intuitive arguments precise.

5.3. Revisiting 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.* [17] 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 polynomial regression algorithm to obtain a the following runtime bound for learning an intersection of k halfspaces under the uniform distribution on $\{-1,1\}^n$. The new bound is better than [17] for $\epsilon > \frac{1}{k}$:

Theorem 8 *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 also be obtained via a boosting-based algorithm similar to one given in recent work due to Jackson et al. [12]. Our approach via the polynomial regression algorithm shows that agnostic learning can have applications even in non-noisy settings.

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

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

[7] K. Clarkson. Subgradient and sampling algorithms for l_1 regression. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 257–266, 2005.

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

[9] 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.

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

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

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

[13] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. Technical Report CUCS-030-05, Columbia University, 2005.

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

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

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

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

[18] 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.

[19] W. S. Lee, P. L. Bartlett, and R. 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.

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

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

[22] 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.

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

[24] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. In *Proceedings of the 24th Annual Symposium on Theory of Computing*, pages 462–467, 1992.

[25] 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.

[26] 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.

[27] 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.

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

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

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

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