Agnostically Learning Boolean Functions with Finite Polynomial Representation*

Ning Ding

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai and State Key Laboratory of Cryptology, Beijing, China
dingning@sjtu.edu.cn

Abstract

Agnostic learning is an extremely hard task in computational learning theory. In this paper we revisit the results in [Kalai et al. SIAM J. Comput. 2008] on agnostically learning boolean functions with finite polynomial representation and those that can be approximated by the former. An example of the former is the class of all boolean low-degree polynomials. For the former, [Kalai et al. SIAM J. Comput. 2008] introduces the l_1-polynomial regression method to learn them to error $\text{opt} + \epsilon$. We present a simple instantiation for one step in the method and accordingly give the analysis. Moreover, we show that even ignoring this step can bring a learning result of error $2\text{opt} + \epsilon$ as well. Then we consider applying the result for learning concept classes that can be approximated by the former to learn richer specific classes. Our result is that the class of s-term DNF formulae can be agnostically learned to error $\text{opt} + \epsilon$ with respect to arbitrary distributions for any ϵ in time $\text{poly}(n^d, 1/\epsilon)$, where $d = O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon))$.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Agnostic Learning, Boolean Functions, Low-Degree Polynomials

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.29

1 Introduction

Learning various boolean function classes plays a central role in computational learning theory. In the PAC learning model [18], a boolean function class $\mathcal{C}$ is learnable if there is an efficient algorithm that, given parameters (ϵ, δ) and many labelled examples of form $(x, f(x))$ where x is chosen from some arbitrary distribution D and $f \in \mathcal{C}$ is an unknown, can with probability $1 - \delta$ output a hypothesis h satisfying $\Pr_{x \leftarrow D}[h(x) \neq f(x)] \leq \epsilon$.

In this model, there are rich boolean function classes that can be learned, such as conjunctions [18], s-term DNF formulas [14], intersections of halfspaces [13], polynomial threshold functions [13, 9] etc. If the underlying distribution D is restricted to some specific ones, some more classes can also be learned. For instance, if D is specified to be the uniform distribution, [15] shows that the Fourier spectrum of any function in AC^0 is concentrated on low-degree coefficients and then introduced the Low Degree Algorithm to learn the low-degree coefficients under the uniform distribution and thus generated a function approximately identical to the concept function. Following [15], some works present various Fourier concentration results for more expressive circuits augmented from AC^0 [10, 2, 7], monotone circuits [3] and boolean functions with small total influence or small noise sensitivity [13] and thus gain corresponding learning results with the Low Degree Algorithm.

* This work is supported by the National Natural Science Foundation of China (Grant No. 61572309) and National Cryptography Development Fund of China (Grant No. MMJJ20170128).

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 29; pp. 29:1–29:11
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Besides the PAC learning model, there is another much harder model, called the agnostic learning model [12, 8]. In this model, a boolean function class $\mathcal{C}$ is learnable if there is an efficient algorithm that, given many pairs of form (x, b) sampled from some arbitrary distribution D, can output a hypothesis f satisfying letting $\mathrm{er}_D(h)$ denote $\Pr_{(x,b)\leftarrow D}[h(x) \neq b]$, $\mathrm{er}_D(f) \leq \mathrm{opt} + \epsilon$, where $\mathrm{opt} = \min_{h\in\mathcal{C}}(\mathrm{er}_D(h))$. So far there have been a few successful attempts to agnostically learning functions. For instance, [11] shows that concept classes that can be approximated by low-degree polynomials can be agnostically learned. Some other works present relaxed requirements for this model: that the output hypothesis f is only required to satisfy $\mathrm{er}_D(f) \leq O(\mathrm{opt}) + \epsilon$ and even at the same time that the learning algorithm only needs to deal with uniform distributions or other specific ones. For instance, [11] shows that boolean function classes with Fourier concentration bounds and halfspaces can be agnostically learned under uniform distributions. [1, 5] show that halfspaces can be agnostically learned to error $O(\mathrm{opt}) + \epsilon$ under isotopic log-concave distributions.

1.1 Our Results

In this paper we revisit the results in [11] on agnostically learning boolean functions with finite polynomial representation and those that can be approximated by the former. By finite polynomial representation, we mean (in an non-rigorous way) that each one in the class admits a polynomial representation in which the number of monomials is much less than 2^n.

More precisely, let $\mathcal{S}$ denote a collection of some subsets of $[n]$. Let $\mathcal{H}_{n,\mathcal{S}}$ denote the class of boolean functions in which each $h(x) = \sum_{S\in\mathcal{S}} g_S \prod_{j\in S} x_j : \{0,1\}^n \to \{0,1\}$ where x_j denotes the jth bit of x and g_S's denote coefficients. Thus $\mathcal{H}_{n,\mathcal{S}}$ is thought of as one with finite polynomial representation if $|\mathcal{S}|$ is not large. For example, $\mathcal{H}_{n,\mathcal{S}}$ is the class of boolean low-degree polynomials if $\mathcal{S}$ consists of all S's with $|S| \leq d$ for some small d.

Recall that [11] presents a result for learning such classes, in which the l_1-polynomial regression method is introduced. Let $p(x)$ denote the polynomial generated by the method. After obtaining $p(x)$, the method outputs $\mathsf{Sign}(p(x) - t)$ for some t as the learned hypothesis. Note that the choice of t is not specified in [11]. So we use a simple sampling technique to determine t. That is, uniformly sample $t \in [0, 1]$ many times and select the one such that $\mathsf{Sign}(p(x) - t)$ is consistent with the most examples. We then show that the t selected this way can indeed satisfy that $\mathsf{Sign}(p(x) - t)$ achieves the error $\mathrm{opt} + \epsilon$. Moreover, we will also show that $t = \frac{1}{2}$ is a universal constant such that for any distribution D, $\mathsf{Sign}(p(x) - \frac{1}{2})$ achieves the error $2\mathrm{opt} + \epsilon$.

Then we consider the question of learning richer classes by applying the general result in [11] for all concept classes admitting low-degree polynomial l_1-approximation in expectation. The concept class in our consideration consists of all s-term DNF formulae. To do this, we show that each s-term DNF formula can be ϵ-uniformly approximated (i.e. l_∞ approximation) by a polynomial of degree $O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon))$. Thus the degree is less than n if $s = O(n^\kappa)$ for any $\kappa < \frac{1}{2}$. Then we have the following result.

▶ **Theorem 1.** *Let D be any distribution over $\{-1,1\}^n \times \{-1,1\}$. For the class of s-term DNF formulae, there is an algorithm that on input (ϵ, δ) and sufficiently many pairs sampled from D can with probability $1 - \delta$ output a hypothesis f such that $er_D(f) \leq opt + \epsilon$ in time $poly(n^d, 1/\epsilon, \log(1/\delta))$ where opt denotes the optimal error among all such DNF formulae and $d = O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon))$.*

Our Techniques. We first outline the technique underlying the first part of this paper. The l_1-polynomial regression method in [11] converts the given examples to a l_1-norm minimization problem. Let f denote the one in $\mathcal{H}_{n,\mathcal{S}}$, achieving the optimal error. For each given pair

$(x, b) \leftarrow D$, b may not equal $f(x)$. So we introduce a variable e to denote $b - f(x)$. Then $e \in \{1, -1, 0\}$. Since $f = \sum_{S \in \mathcal{S}} g_S \prod_{j \in S} x_j$, substituting the value of x_j into f and letting $a_S = \prod_{j \in S} x_j$, we obtain $\sum_{S \in \mathcal{S}} g_S a_S + e = b$. Viewing all a_S's as coefficients, this equality is a linear equation of all g_S's. We also use $\mathbf{a}$ to denote the (row) vector $(a_{S_1}, \cdots, a_{S_N})$ (where we assume there is an order for all sets in $\mathcal{S}$ and let $N = |\mathcal{S}|$). Let $\mathbf{g}$ denote the (column) vector $(g_{S_1}, \cdots, g_{S_N})$. Thus the equation is $\mathbf{a} \cdot \mathbf{g} + e = b$.

Thus when given m random pairs, we can construct m equations of form $\mathbf{a} \cdot \mathbf{g} + e = b$. Let $\mathbf{A}$ denote the $m \times N$ matrix consists of all such $\mathbf{a}$ as rows, $\mathbf{e}$ denote the (column) vector consisting of all e's, $\mathbf{b}$ denote the (column) vector consisting of all b's. Thus m equalities can be represented as $\mathbf{A} \cdot \mathbf{g} + \mathbf{e} = \mathbf{b}$. Then the l_1-polynomial regression method finds a solution $\mathbf{g}$ such that $\mathbf{A} \cdot \mathbf{g} - \mathbf{b}$ achieves the minimal l_1-norm. Let $p(x)$ denote the polynomial formed using $\mathbf{g}$. After obtaining $p(x)$, the method outputs $\mathsf{Sign}(p(x) - t)$ for some t as the learned hypothesis.

Note that the choice of t is not specified in [11]. So we consider using uniformly sampled t. That is, uniformly sample $t \in [0, 1]$ many times and select the one such that $\mathsf{Sign}(p(x) - t)$ is consistent with the most examples. We then show that the t selected this way can indeed satisfy that $\mathsf{Sign}(p(x) - t)$ achieves the error $\text{opt} + \epsilon$. Moreover, we will show that due to the l_1-polynomial strategy, there is at most 2opt-fraction of the examples such that $|p(x) - b| \geq \frac{1}{2}$, which means that there is at least $1 - 2\text{opt}$ fraction such that $|p(x) - b| < \frac{1}{2}$. Thus $\mathsf{Sign}(p(x) - \frac{1}{2})$ is correct on this $1 - 2\text{opt}$ fraction of the examples. This shows that $t = \frac{1}{2}$ is a universal constant such that $\mathsf{Sign}(p(x) - \frac{1}{2})$ achieves the error $2\text{opt} + \epsilon$.

Then we sketch the technique underlying the second part. By using the uniform approximations for OR and AND operations in [17] twice, we show that each s-term DNF formula f can be ϵ-uniformly approximated by a polynomial p of degree $O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon))$. This ensures that the expectation of $|f - p|$ is less than ϵ. Then applying the general result in [11], we obtain the learning result for s-term DNF formulae.

1.2 Organization

The rest of this paper is arranged as follows. Section 2 presents the preliminaries used throughout the paper. Section 3 recalls the l_1-polynomial regression method in [11] in which we instantiate the choice of t and show the universality of $\frac{1}{2}$. Section 4 presents the result for learning s-term DNF formulae.

2 Preliminaries

This section contains the notations and definitions used throughout this paper.

2.1 Basic Notions

Let $[n]$ denote the integers in $[1, n]$. Let $\mathbf{Z}, \mathbf{Q}, \mathbf{R}$ denote integers, rational numbers and reals. For any vector $\mathbf{z} = (z_1, \cdots, z_m) \in \mathbf{R}^m$, $\|\mathbf{z}\|_1$ denotes its l_1-norm, defined as $\sum_{i=1}^m |z_i|$. For a vector $\mathbf{v} \in \mathbf{R}^m$ and a set $I \subset [m]$, we denote by $\mathbf{v}_I$ the vector in $\mathbf{R}^m$ which coincides with $\mathbf{v}$ on the indices in I and is extended to zero outside I. We say that a vector $\mathbf{e} \in \mathbf{R}^m$ is s-sparse if the number of non-zero entries of $\mathbf{e}$ is at most s.

Let $\lfloor \cdot \rceil$ denote the operation of rounding to the nearest integer.

For any distribution D over $\{0, 1\}^n \times \{0, 1\}$, letting D's output be of form (x, b), we will use (x^k, b_k) to denote the output of D in the kth sampling, while we use x_j to denote the jth bit of x, $1 \leq j \leq n$.

Let $(x^1, b_1), \cdots, (x^m, b_m)$ denote m pairs drawn from D independently. We say a function f is consistent with α fraction of the pairs if $|\{k \in [m] : f(x^k) = b_k\}|/m = \alpha$. Following literatures, we say f is consistent with the pairs if $\alpha = 1$ and say it is approximate-consistent if $0 < \alpha < 1$ which differs from 1 by a small quantity.

Let $\mathsf{Sign}(\cdot)$ denote the function that on input y outputs 1 if $y \geq 0$ and outputs 0 otherwise.

For a boolean function class H, and a set S of M points in the input space X, if the restriction of H to the set S computes all 2^M functions on S, we say that H shatters S. The VC-dimension of H is the size of the largest shattered subset of X, also denoted $\mathrm{VCdim}(H)$.

2.2 Agnostic Learning

Informally, in the agnostic learning model [12, 8], there is a class of functions $\mathcal{C}$ which we wish to learn. We consider each function of $\mathcal{C}$ is boolean. Each example-label pair is chosen from a distribution D over $X \times \{0, 1\}$ (X denotes the input space). When given many pairs, the learning algorithm is supposed to output a function f that can achieve almost the minimal error among all functions in $\mathcal{C}$ with respect to D.

For any function f, let $\mathrm{er}_D(f)$ denote $\Pr_{(x,b)\leftarrow D}[f(x) \neq b]$. A training sample drawn from D is of form $((x^1, b_1, \cdots, (x^m, b_m))$ where each (x^k, b_k) is drawn from D independently $1 \leq k \leq m$.

▶ **Definition 2.** (Agnostic Learning). Let D be a distribution on $X \times \{0, 1\}$ and let $\mathcal{C}$ be a class of boolean functions. We say that an algorithm L agnostically learns $\mathcal{C}$ if L is given (ϵ, δ) and many random example-label pairs drawn from any D, then with probability $1 - \delta$, L outputs a hypothesis f such that $\mathrm{er}_D(f) \leq \mathrm{opt} + \epsilon$, where opt denotes $\min_{h\in\mathcal{C}}(\mathrm{er}_D(h))$.

If L can only work under some specific distribution D, we say L agnostically learns $\mathcal{C}$ under D. We refer to ϵ as the accuracy parameter and δ as the confidence parameter.

We also consider a relaxation by only requiring that the f output by L is such that $\mathrm{er}_D(f) \leq O(\mathrm{opt}) + \epsilon$.

The learning algorithm sometimes needs some additional input parameters. For instance, the Low Degree algorithm has as input the maximal Fourier degree. For our learning algorithm for $\mathcal{H}_{n,\mathcal{S}}$ in this paper, it needs to have as input some representation of $\mathcal{S}$.

3 On Learning Boolean Polynomials

In this section we revisit the result of learning boolean polynomials in [11], in which the l_1-polynomial regression method is employed. We recall this method, instantiate one strategy in it and accordingly present the analysis. Moreover, we show that even ignoring this strategy can bring a learning result of error $2\mathrm{opt} + \epsilon$ as well. In Section 3.1 we demonstrate this learning task and introduce the notations. In Section 3.2 we present the the instantiation and analysis for the l_1-polynomial regression method to find hypotheses consistent with given examples. In Section 3.3 we follow the standard way to convert consistent-hypotheses to learned hypotheses.

3.1 Goal and Notations

Let $h : \{0, 1\}^n \to \{0, 1\}$ be any one in $\mathcal{H}_{n,\mathcal{S}}$, which can be represented as $h(x) = \sum_{S\in\mathcal{S}} g_S \prod_{j\in S} x_j$ over $x_1, \cdots, x_n$, where g_S's denote the coefficients. So the task of learning $\mathcal{H}_{n,\mathcal{S}}$ is to output a boolean function f' (not necessarily in $\mathcal{H}_{n,\mathcal{S}}$) when given many pairs of form (x, b) sampled from any distribution D over $\{0, 1\}^n \times \{0, 1\}$, such that f' achieves almost the optimal error among all ones in $\mathcal{H}_{n,\mathcal{S}}$. Typically, if $\mathcal{S}$ consists of all S's with $|S| \leq d$, the task is actually the agnostic learning of boolean d-degree polynomials.

Precisely, let $(x^1, b_1), \cdots, (x^m, b_m)$ denote m pairs independently sampled from D. Then the learning goal is, when given (ϵ, δ), with probability $1-\delta$, to output a hypothesis f' satisfying $\Pr[f'(x) \neq b] \leq \text{opt} + \epsilon$ for $(x, b) \leftarrow D$, where $\text{opt} = \min_{h \in \mathcal{H}_{n,\mathcal{S}}}(\Pr_{(x,b)\leftarrow D}[h(x) \neq b])$.

Assume that $f \in \mathcal{H}_{n,\mathcal{S}}$ is the one satisfying $\text{opt} = \text{er}_D(f)$. For each pair (x^k, b_k), we view b_k as the sum of $f(x^k)$ and an error e_k. That is, $b_k = f(x^k) + e_k$. Thus, each e_k is of value in $\{0, -1, 1\}$, in which $e_k = 0$ indicates $f(x^k) = b_k$ and $e_k = \pm 1$ indicates $f(x^k) = 1 - b_k$. Let x_j^k denote the jth bit of x^k. For (x^k, b^k), we can generate an equality as follows.

$$\sum_{S \in \mathcal{S}} g_S \prod_{j \in S} x_j^k + e_k = b_k, k \in [1, m]$$

Let a_S^k be the value of $\prod_{j \in S} x_j^k$. Then list the m equalities as follows.

$$\begin{cases} \sum_{S \in \mathcal{S}} g_S a_S^1 + e_1 = b_1 \\ \cdots\cdots\cdots\cdots\cdots \\ \sum_{S \in \mathcal{S}} g_S a_S^m + e_m = b_m \end{cases} \tag{1}$$

In the above equalities, all g_S's are unknown and the goal of learning is to recover them. Viewing all a_S^k as coefficients, the equalities are linear for the unknown variables g_S's. For convenience, for all $S \in \mathcal{S}$, we use $S_1, \cdots, S_N$ denote all of them where $N = |\mathcal{S}|$.

Let $\mathbf{a}^k$ denote the (row) vector $(a_{S_1}^k, \cdots, a_{S_N}^k) \in \mathbf{Z}^N$. Let $\mathbf{g}$ denote the (column) vector $(g_{S_1}, \cdots, g_{S_N}) \in \mathbf{Z}^N$. Then for the kth example, we have

$$\mathbf{a}^k \cdot \mathbf{g} + e_k = b_k$$

Let $\mathbf{e}$ denote the (column) vector $(e_1, \cdots, e_m) \in \mathbf{Z}^m$. Let $\mathbf{A}$ denote the m by N matrix which rows consist of all $\mathbf{a}^k$'s. Let $\mathbf{b}$ denote the (column) vector $(b_1, \cdots, b_m) \in \mathbf{Z}^m$. Then the m linear equations can be written as

$$\mathbf{A} \cdot \mathbf{g} + \mathbf{e} = \mathbf{b}$$

Then we can define the following problem: find a solution $\mathbf{g}^*$ such that

$$\|\mathbf{A} \cdot \mathbf{g}^* - \mathbf{b}\|_1 = \inf_{\mathbf{g}'} \|\mathbf{A} \cdot \mathbf{g}' - \mathbf{b}\|_1$$

where $\mathbf{g}', \mathbf{g}^*$ should satisfy that each entry of $\mathbf{A} \cdot \mathbf{g}'$ and $\mathbf{A} \cdot \mathbf{g}^*$ is in $[0, 1]$. This problem can be solved using linear programming.

When obtaining a solution $\mathbf{g}^*$, let $\mathbf{z}$ denote $\mathbf{b} - \mathbf{A}\mathbf{g}^*$. Then we can run the remaining strategy of the l_1-polynomial regression to generate a consistent-hypothesis as well as a learned one. In the rest of this section we will formalize these procedures.

3.2 Finding Consistent-Hypotheses

Recall that $(x^1, b_1), \cdots, (x^m, b_m)$ denote m pairs sampled from D independently, $1 \leq k \leq m$, and f is the function in $\mathcal{H}_{n,\mathcal{S}}$ which achieves opt-error with respect to D. Refer to Section 3.1 for the definitions of notations $\mathbf{A}, \mathbf{b}, \mathbf{g}^*, \mathbf{e}, \mathbf{z}$.

Algorithm 1: The consistent-hypothesis-finder.

Input:
- m pairs of form (x, b) drawn from D independently.
- ϵ, δ and the knowledge of $\mathcal{S}$.

Output: a hypothesis f_0.

1. Run a l_1-polynomial regression algorithm to find a solution $\mathbf{g}^*$ such that
$$\|\mathbf{A} \cdot \mathbf{g}^* - \mathbf{b}\|_1 = \inf_{\mathbf{g}'} \|\mathbf{A} \cdot \mathbf{g}' - \mathbf{b}\|_1$$
where $\mathbf{g}', \mathbf{g}^*$ satisfy that each entry of $\mathbf{A} \cdot \mathbf{g}', \mathbf{A} \cdot \mathbf{g}^*$ is in $[0, 1]$.
Assume that $\mathbf{g}^*$ consists of all g_S^*'s. Let $p(x) = \sum_{S \in \mathcal{S}} g_S^* \prod_{j \in S} x_j$. (Thus $p(x^k) \in [0, 1]$ for $1 \leq k \leq m$.)
2. Uniformly sample $t \in (0, 1)$ $O(1 + 1/\epsilon) \ln(\frac{1}{\delta})$ times. Select one t satisfying $f_0(x) = \mathsf{Sign}(p(x) - t)$ achieves the minimal empirical error on the m examples and finally output f_0.

End Algorithm

Let I denote the set of the indices $k \in [m]$ on which $e_k \neq 0$. Let $\mu = |I|/m$. (It can be seen that $\mu \approx$ opt.)

First it can be seen that since $\mathbf{e} = \mathbf{b} - \mathbf{Ag}$ and $\mathbf{g}^*$ achieves the minimal $\|\mathbf{b} - \mathbf{Ag}^*\|_1$ among all $\mathbf{g}'$, $\|\mathbf{z}\|_1 \leq \|\mathbf{e}\|_1 = |I|$. Then we follow the method of [11] to construct a consistent hypothesis as shown in Algorithm 1, in which we instantiate the second step for determining t.

For distribution D, let $\mathrm{er}_D(h)$ denote $\Pr[h(x) \neq b]$ for $(x, b) \leftarrow D$. Let Z denote pairs $(x^1, b_1), \cdots, (x^m, b_m)$. Then let $\widehat{\mathrm{er}}_Z(h)$ denote $\frac{1}{m}|\{k : h(x^k) \neq b_k\}|$.

▶ **Proposition 3.** *With probability* $1 - \delta$*, the hypothesis* $f_0(x)$ *in Algorithm 1 is such that* $\widehat{\mathrm{er}}_Z(f_0) \leq \mu + \mu\epsilon < \mu + \epsilon$.

Proof. Let h denote $\mathsf{Sign}(p(x) - t)$ for uniform t. First using the argument of [11] (the proof of Theorem 5), we have the following claim.

$$\mathbf{E}_t[\widehat{\mathrm{er}}_Z(h)] \leq \frac{1}{m} \sum_{k=1}^{m} |p(x^k) - b^k|$$

To see this, $\mathbf{E}_t[\widehat{\mathrm{er}}_Z(h)]$ equals the average sum of the probabilities of all events $h(x^k) \neq b^k$. Thus for each (x^k, b^k), $f_0(x^k) \neq b^k$ if t lies between $p(x^k)$ and b^k. Note that $p(x^k) \in [0, 1]$ and $b^k \in \{0, 1\}$. Hence, for uniform $u \in (0, 1)$, for any k, the probability that t lies in between the two numbers is $|p(x^k) - b^k|$. So the above inequality holds.

Then notice that
$$\frac{1}{m} \sum_{k=1}^{m} |p(x^k) - b^k)| = \frac{1}{m} \sum_{k=1}^{m} |z_k| = \frac{1}{m} \cdot \|\mathbf{z}\|_1 \leq \frac{1}{m} \cdot \|\mathbf{e}\|_1 = \frac{|I|}{m} = \mu$$

So $\mathbf{E}_t[\widehat{\mathrm{er}}_Z(h)] \leq \mu$. Furthermore, by Markov's inequality, $\Pr[\widehat{\mathrm{er}}_Z(h) > (1+\epsilon)\mu] \leq \frac{\mu}{(1+\epsilon)\mu} = \frac{1}{1+\epsilon} = 1 - \frac{\epsilon}{1+\epsilon}$. Thus

$$\Pr[\widehat{\mathrm{er}}_Z(h) \leq (1 + \epsilon)\mu] > \frac{\epsilon}{1 + \epsilon}$$

So for $O(1+1/\epsilon) \ln(\frac{1}{\delta})$ times sampling of u, with probability $1-(1-\frac{\epsilon}{1+\epsilon})^{O(1+1/\epsilon) \ln \frac{1}{\delta}} > 1-\delta$, there is at least one u such that $\widehat{\mathrm{er}}_Z(h) \leq (1+\epsilon)\mu < \mu+\epsilon$. Then f_0 is this h. The proposition holds. ◀

We remark that Proposition 3 can be extended to any concept class $\mathcal{C}$ that can be l_1- (or l_2) approximated by $\mathcal{H}_{n,\mathcal{S}}$ in expectation as shown [11].

In the following we show that $t = \frac{1}{2}$ is a universal constant such that for any distribution D, letting $f_0 = \mathsf{Sign}(p(x) - \frac{1}{2})$ in Algorithm 1 (ignoring (ϵ, δ) and omitting the second step), the following result holds.

▶ **Proposition 4.** *The hypothesis $f_0(x) = \mathsf{Sign}(p(x) - \frac{1}{2})$ is such that $\widehat{er}_Z(f_0) \leq 2\mu$.*

Proof. Notice that $\mathbf{A} \cdot \mathbf{g}^* = \mathbf{b} - \mathbf{z}$. First since $\|\mathbf{z}\|_1 \leq \|\mathbf{e}\|_1 = \mu m$, there is at most 2μ fraction of $k \in [1, m]$ such that $|z_k| \geq \frac{1}{2}$. That is, there is at least $1 - 2\mu$ fraction of all k's satisfying $|z_k| < \frac{1}{2}$. This means that for this $1 - 2\mu$ fraction of all k's, $p(x^k)$ differs from b^k by a quantity less than $\frac{1}{2}$. This also means that $\lfloor p(x^k) \rceil$ equals b^k for this fraction. We now show this rounding to the closest integers is identical to the sign operation to $p(x^k) - \frac{1}{2}$ for this fraction. It can be seen that if $b^k = 1$, $p(x^k)$ is more than $\frac{1}{2}$. Thus $\lfloor p(x^k) \rceil$ will output 1. In this case $\mathsf{Sign}(p(x^k) - \frac{1}{2})$ outputs 1 either. If $b^k = 0$, $p(x^k)$ is less than $\frac{1}{2}$. Thus $\lfloor p(x^k) \rceil$ will output 0. In this case $\mathsf{Sign}(p(x^k) - \frac{1}{2})$ outputs 0 either. The proposition holds. ◀

3.3 The Learning Result

In the rest of this section we present the required sample complexity and state the learning result. Let $\mathcal{F}_{n,\mathcal{S}}$ denote the boolean function class, in which each one on input $x \in \{0,1\}^n$ first computes $\prod_{j \in S} x_j$ for all $S \in \mathcal{S}$ and compute a halfspace of all $\prod_{j \in S} x_j$. Thus it can be seen that $\mathcal{H}_{n,\mathcal{S}}$ and the output hypotheses of Algorithm 1 are in $\mathcal{F}_{n,\mathcal{S}}$. In the following let us estimate the VC-dimension of $\mathcal{F}_{n,\mathcal{S}}$.

▶ **Proposition 5.** *$\mathcal{F}_{n,\mathcal{S}}$ is contained in the class of 2-level threshold circuits of $|\mathcal{S}| \cdot (n+1)$ weights and thresholds and $|\mathcal{S} + 1|$ computation gates which is of VC-dimension $O(n \cdot |\mathcal{S}| \cdot \log |\mathcal{S}|)$.*

Proof. First each monomial of form $\prod_{j \in S} x_j$ can be computed by an AND gate of $j \leq n$ inputs and each AND gate of n inputs can be computed by a threshold gate of the n inputs and $n + 1$ weights and threshold. Thus f can be computed by a 2-level threshold circuits in which the first level computes $\prod_{j \in S} x_j$ for all $S \in \mathcal{S}$ and the second computes the threshold gate above. It can be seen that this circuit is of $O(|\mathcal{S}| \cdot n)$ weights and thresholds and $|\mathcal{S}| + 1$ gates in total. Thus due to [4], the VC dimension of all such circuits is $O(n \cdot |\mathcal{S}| \cdot \log |\mathcal{S}|)$. ◀

Then recall the following result.

▶ **Theorem 6.** *([19]) Let D be any distribution over $\{0,1\}^n \times \{0,1\}$. Let Z denote m pairs independently sampled from D. For $0 < \epsilon < 1$, it holds that for all $h \in \mathcal{F}_{n,\mathcal{S}}$,*

$$\Pr[|er_D(h) - \widehat{er}_Z(h)| \geq \epsilon] \leq \delta, \quad \textit{if } m \geq \frac{64}{\epsilon^2}(2\,VCdim(\mathcal{F}_{n,\mathcal{S}}) \ln(\frac{12}{\epsilon}) + \ln(\frac{4}{\delta}))$$

Suppose that when given Z, $f_0 \in \mathcal{F}_{n,\mathcal{S}}$ is a hypothesis such that $\mathrm{er}_Z(f_0) \leq c \cdot \mathrm{opt} + \epsilon_0$ for some constant c ($c = 1$ in Proposition 3 and $c = 2$ in Proposition 4). Then we have the following result.

▶ **Claim 7.** *When $m \geq \frac{64}{\epsilon^2}(2\,VCdim(\mathcal{F}_{n,\mathcal{S}}) \ln(\frac{12}{\epsilon}) + \ln(\frac{4}{\delta}))$ and let Z, D, f_0 be defined as above, with probability $1 - \delta$, $er_D(f_0) < c \cdot opt + \epsilon_0 + \epsilon$.*

Proof. Given the condition of m, by Theorem 6, we have that with probability $1 - \delta$, $|\mathrm{er}_D(h) - \widehat{\mathrm{er}}_Z(h)| \leq \epsilon$ for all $h \in \mathcal{F}_{n,\mathcal{S}}$. Thus for $f_0 \in \mathcal{F}_{n,\mathcal{S}}$, we have

$$\mathrm{er}_D(f_0) \leq \widehat{\mathrm{er}}_Z(f_0) + \epsilon \leq c \cdot \mathrm{opt} + \epsilon_0 + \epsilon$$

The claim holds. ◀

Combining Proposition 5 and Claim 7, we have the following proposition.

▶ **Proposition 8.** *Choosing* $m \geq O(\frac{1}{\epsilon^2}(n|\mathcal{S}|\log|\mathcal{S}|\ln(\frac{12}{\epsilon})+\ln(\frac{4}{\delta})))$ *and letting* $f_0 \in \mathcal{F}_{n,\mathcal{S}}$ *be such that* $\widehat{er}_Z(f_0) < c \cdot opt + \epsilon_0$ *where* Z *denotes* m *pairs sampled from* D*, with probability* $1-\delta$*,* $er_D(f_0) < c \cdot opt + \epsilon_0 + \epsilon$*.*

Then we estimate $|I|$ as follows, which will be used in the proof of Proposition 10.

▶ **Claim 9.** *For any* $0 < \delta < 1$*, with probability* $1-\delta$*,* $|I| \leq (opt \cdot m + \sqrt{3\ln\frac{1}{\delta} \cdot opt \cdot m})$*.*

Proof. Let $\xi_k = 1$ if $e_k \neq 0$ and $\xi_k = 0$ if $e_k = 0$ for $1 \leq k \leq m$. Let $X = \sum_{k=1}^m \xi_k$. Then $\mathbf{E}[X] = \mathrm{opt} \cdot m$. Due to the Chernoff bound, for any $0 < \lambda < 1$,

$$\Pr[X < (1+\lambda)\mathbf{E}[X]] > 1 - e^{-\lambda^2\mathbf{E}[X]/3}$$

So set $\lambda = \sqrt{3\ln\frac{1}{\delta}} \cdot \frac{1}{\sqrt{\mathrm{opt}\cdot m}}$. Then the above probability formula is simplified to

$$\Pr[X < (\mathrm{opt} \cdot m + \sqrt{3\log\frac{1}{\delta} \cdot \mathrm{opt} \cdot m})] > 1 - \delta$$

The claim holds. ◀

Lastly, replace ϵ, δ in Algorithm 1 by $\frac{\epsilon}{3}, \frac{\delta}{3}$. We have the following result.

▶ **Proposition 10.** *Algorithm 1 can with probability at least* $1-\delta$ *output a hypothesis, denoted* f_0 *in time poly*$(|\mathcal{S}|, n, \frac{1}{\epsilon}, \log\frac{1}{\delta})$ *satisfying* $er_D(f_0) \leq c \cdot opt + \epsilon$*, where* $opt = \min_{h\in\mathcal{H}_{n,\mathcal{S}}}(er_D(h))$ *(*$c = 1$ *when using Proposition 3 or* $c = 2$ *when using Proposition 4).*

Proof. By Claim 9, except for probability $\frac{\delta}{3}$, $\mu = |I|/m \leq \mathrm{opt} + \sqrt{3\ln\frac{3}{\delta} \cdot \mathrm{opt}} \cdot m^{-\frac{1}{2}}$. By Proposition 3 (or Proposition 4), except for another $\delta/3$ probability, $\widehat{\mathrm{er}}_Z(f_0) \leq c\mu + \epsilon/3 = c \cdot \mathrm{opt} + O(m^{-1/2}) + \epsilon/3$, where Z denotes the sample consisting of the m pairs. So by Proposition 8, $\mathrm{er}_D(f_0) \leq c \cdot \mathrm{opt} + O(m^{-1/2}) + 2\epsilon/3 < c \cdot \mathrm{opt} + \epsilon$ (where $O(m^{-1/2}) < \epsilon/3$), and the total failure probability is at most δ.

Moreover, $(\mathbf{A}, \mathbf{b})$ can be generated in time polynomial in $(|\mathcal{S}|, m)$, and the l_1-polynomial regression algorithm runs in time polynomial in its input. Thus the time complexity holds. ◀

4 Learning DNF Formulae

In this section we present an agnostic learning result for DNF formulae, as an application of the general result in [11] for all concept classes admitting l_1-approximation with low-degree polynomials in expectation. Recall that s-term DNF formulae can be PAC learned in time $n^{O(n^{1/3}\cdot\log s)}$ [13], and [6] combined with [16] presents a query algorithm to agnostically learn DNFs in time $n^{O(\log(1/\epsilon)\log\log n)}$ under the uniform distribution. We will present an agnostically learning algorithm for s-term DNF formulae ($s < \sqrt{n}$) by showing that such DNF formulae have uniform approximation with low-degree polynomials. First, let us recall the general result in [11] as follows.

▶ **Theorem 11.** *([11]) Let* $\mathcal{C}$ *denote a concept class,* D *be any distribution over* $\{-1,1\}^n \times \{-1,1\}$*. Assume for any hypothesis* $h \in \mathcal{C}$*, there is a polynomial* p *of degree* d *such that* $\mathbf{E}_D[|h(x) - p(x)|] < \epsilon$*. Then there is an algorithm that on input parameters* (ϵ, δ) *and* d*, sufficiently many pairs sampled from* D *independently can with probability* $1-\delta$ *output a hypothesis* f *such that* $er_D(f) \leq opt + \epsilon$ *in time poly*$(n^d, \frac{1}{\epsilon}, \log\frac{1}{\delta})$ *where* $opt = \min_{h\in\mathcal{C}}(er_D(h))$*.*

Let f be a boolean function mapping $\{-1,1\}^n \to \{-1,1\}$. Let p be a degree-$d_{\epsilon'}$ n-variate polynomial mapping $\mathbf{R}^n \to \mathbf{R}$. We say that $p(x)$ ϵ'-uniformly approximates $f(x)$ if $|f(x)-p(x)| \le \epsilon'$ for any $x \in \{-1,1\}^n$.

In the following we show that each s-term DNF formula f can be $2\epsilon'$-uniformly approximated by a polynomial p of degree $O(\sqrt{n}\cdot s\cdot \log s \log^2(1/\epsilon'))$ for any ϵ'. This implies $\mathbf{E}_D[|f-p|] \le 2\epsilon'$. Thus by Theorem 11 we obtain the result of agnostically learning s-term DNFs.

Now consider f as a DNF formula which is the OR of s conjunctions $f_1,\cdots,f_s$. W.l.o.g., assume each f_i is the AND of at most n literals in $\{x_1,\cdots,x_n,\overline{x}_1,\cdots,\overline{x}_n\}$. (If it connects more than n literals, then there is an j such that $x_j,\overline{x}_j$ appear in it simultaneously, which means it is always equal to false and thus can be got rid of from f.) In the following we show that f admits a uniform approximation.

▶ **Proposition 12.** *Each s-term DNF formula f can be $2\epsilon'$-uniformly approximated by a polynomial p of degree $O(\sqrt{n}\cdot s\cdot \log s \log^2(1/\epsilon'))$ for any ϵ'.*

Proof. By [17], for each AND of n variables, for any ϵ_0, there is a multi-variate real polynomial that can ϵ_0-uniformly approximate it. That is, for each f_i, there is a $p_i(x)$ of degree $O(\sqrt{n}\log(1/\epsilon_0))$ satisfying $|p_i(x)-f_i(x)| \le \epsilon_0$ for all $x \in \{-1,1\}^n$. It can be seen that $f_i(x)/p_i(x) \in [\frac{1}{1+\epsilon_0}, \frac{1}{1-\epsilon_0}]$ for $f_i(x) = \pm 1$ and for each i.

Notice that $\frac{1}{1+\epsilon_0} > 1-\epsilon_0$. Since $(1-\epsilon_0)(1+2\epsilon_0) = 1+\epsilon_0-2\epsilon_0^2$, choosing $\epsilon_0 < \frac{1}{n^2}$, we have that

$$\frac{1}{1-\epsilon_0} = \frac{1+2\epsilon_0}{1+\epsilon_0-2\epsilon_0^2} < 1+2\epsilon_0$$

So $f_i(x)/p_i(x) \in (1-\epsilon_0, 1+2\epsilon_0)$ for all i's. Let $f_i(x)/p_i(x) = 1+\Delta_i(x)$. Then $\Delta_i \in (-\epsilon_0, 2\epsilon_0)$.

Since f is OR of $f_1,\cdots,f_s$, using [17] again, we have that there exists an s-variate multi-linear polynomial $P(f_1,\cdots,f_s)$ of degree $O(\sqrt{s}\log(1/\epsilon'))$ such that $|f(f_1,\cdots,f_s) - P(f_1,\cdots,f_s)| \le \epsilon'$ for any $f_1,\cdots,f_s$. Denote the Fourier expansion of $P(f_1,\cdots,f_s)$ by $\sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S} f_j$, where each $S \subset [n]$ and β_S's are coefficients each of which is less than a constant. Thus we have

$$\begin{aligned}
P(f_1,\cdots,f_s) &= \sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S} f_j = \sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S}(p_j\cdot(1+\Delta_j)) \\
&= \sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S} p_j \cdot \prod_{j\in S}(1+\Delta_j) \\
&= \sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S} p_j \cdot (1+\sum_{j=1}^{|S|}\Delta_j + O(\max_j(\Delta_j))) \\
&= P(p_1,\cdots,p_s) + \sum_{|S|\le O(\sqrt{s}\log(1/\epsilon'))} \beta_S \prod_{j\in S} p_j (\sum_{j=1}^{|S|}\Delta_j + O(\max_j(\Delta_j)))
\end{aligned}$$

When $\epsilon_0 \cdot n \cdot \binom{s}{O(\sqrt{s}\log(1/\epsilon'))} < \epsilon'/n$, the second addend in the right side of the last equality is less than ϵ'. Thus in the beginning, we would choose

$$\epsilon_0 < \frac{\epsilon'}{n^2}\cdot s^{-O(\sqrt{s})\log(1/\epsilon')}$$

Then each $p_i(x)$ is of degree $O(\sqrt{n}\log(1/\epsilon_0)) = O(\sqrt{n}\cdot(\log n + \sqrt{s}\log s\log(1/\epsilon'))) = O(\sqrt{ns}\log s\log(1/\epsilon'))$ (when $\sqrt{s} > \log n$).

More importantly, we have $|P(f_1, \cdots, f_s) - P(p_1, \cdots, p_s)| < \epsilon'$, which shows that $|f(f_1(x), \cdots, f_s(x)) - P(p_1(x), \cdots, p_s(x))| < 2\epsilon'$ for any $x \in \{-1, 1\}^n$.

Notice that $P(p_1(x), \cdots, p_s(x))$ is actually a multi-linear polynomial on x of degree $O(\sqrt{ns} \log s \log(1/\epsilon')) \cdot O(\sqrt{s} \log(1/\epsilon')) = O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon'))$. The proposition holds. ◀

Thus we have the following learning result.

▶ **Proposition 13.** *For each s, for any (ϵ, δ), all s-term DNF formulae can be agnostically learned to error $opt + \epsilon$ and confidence δ in time poly$(n^d, \frac{1}{\epsilon}, \log \frac{1}{\delta})$, where $d = O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon))$.*

Proof. By Proposition 12, $\mathbf{E}_D[|f(x) - P(p_1(x), \cdots, p_s(x))|] \leq 2\epsilon'$ for any $\epsilon' > 0$ where $P(p_1(x), \cdots, p_s(x)$ is of degree $O(\sqrt{n} \cdot s \cdot \log s \log^2(1/\epsilon'))$. Thus, choosing $\epsilon = 2\epsilon'$, by Theorem 11, the proposition holds. ◀

Acknowledgements. The author is grateful to the reviewers of ISAAC 2017 for their useful comments.

References

1 Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 449–458. ACM, 2014. doi:10.1145/2591796.2591839.

2 Richard Beigel. When do extra majority gates help? polylog(n) majority gates are equivalent to one. *Computational Complexity*, 4:314–324, 1994.

3 Nader H. Bshouty and Christino Tamon. On the fourier spectrum of monotone functions. *J. ACM*, 43(4):747–770, 1996. doi:10.1145/234533.234564.

4 T. M. Cover. Capacity problems for linear machines. *Patten Recognition*, pages 283–289, 1968.

5 Amit Daniely. A PTAS for agnostically learning halfspaces. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 484–502. JMLR.org, 2015. URL: http://jmlr.org/proceedings/papers/v40/Daniely15.html.

6 Parikshit Gopalan, Adam Tauman Kalai, and Adam R. Klivans. Agnostically learning decision trees. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 527–536. ACM, 2008. doi:10.1145/1374376.1374451.

7 Parikshit Gopalan and Rocco A. Servedio. Learning and lower bounds for ac^0 with threshold gates. In Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 6302 of *Lecture Notes in Computer Science*, pages 588–601. Springer, 2010.

8 David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

9 Lisa Hellerstein and Rocco A. Servedio. On PAC learning algorithms for rich boolean function classes. *Theor. Comput. Sci.*, 384(1):66–76, 2007. doi:10.1016/j.tcs.2007.05.018.

10 Jeffrey C. Jackson, Adam Klivans, and Rocco A. Servedio. Learnability beyond ac0. In *IEEE Conference on Computational Complexity*, page 26. IEEE Computer Society, 2002.

11 Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008. doi:10.1137/060649057.

12 Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

13 Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *J. Comput. Syst. Sci.*, 68(4):808–840, 2004. doi:10.1016/j.jcss.2003.11.002.

14 Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{\tilde{o}(n^{1/3})}$. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 258–265. ACM, 2001. doi:10.1145/380752.380809.

15 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

16 Yishay Mansour. An o(nˆ(log log n)) learning algorithm for DNT under the uniform distribution. *J. Comput. Syst. Sci.*, 50(3):543–550, 1995. doi:10.1006/jcss.1995.1043.

17 Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994. doi:10.1007/BF01263419.

18 Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

19 V.N.Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.