An $O(n^{\log\log n})$ Learning Algorithm for DNF under the Uniform Distribution

Yishay Mansour*

February 4, 1998

Abstract

We show that a DNF with terms of size at most d can be approximated by a function with at most $d^{O(d \log 1/\varepsilon)}$ non zero Fourier coefficients such that the expected error squared, with respect to the uniform distribution, is at most ε. This property is used to derive a learning algorithm for DNF, under the uniform distribution.

The learning algorithm uses queries and learns, with respect to the uniform distribution, a DNF with terms of size at most d in time polynomial in n and $d^{O(d \log 1/\varepsilon)}$. The interesting implications are for the case when ε is constant. In this case our algorithm learns a DNF with a polynomial number of terms in time $n^{O(\log\log n)}$, and a DNF with terms of size at most $O(\log n / \log\log n)$ in polynomial time.

1 INTRODUCTION

One of the most basic problems in theoretical machine learning has been learning the class of DNFs with a polynomial number of terms. This task has been an open problem since the work of Valiant [Val84] introducing the PAC model. Learning a DNF with a polynomial number of terms in polynomial time remains an open question even when the examples are drawn from the uniform distribution and the learner is allowed to query the DNF.

In his seminal paper Valiant [Val84] gave a polynomial time algorithm for learning k-CNF, in the PAC model, and showed how to learn polynomial size monotone DNF formulas using queries. There has been some success in devising algorithms for learning DNF with various restriction on the number of times a variable can appear in the DNF formula (see [KLPV87, Han91, HM91, AP91]). Unfortunately none of the results seem to extend to the general case.

Negative results have been shown for learning DNF in the PAC model. In [PV88] it was shown that deciding if a given set of examples can be described as a two term DNF is NP-Complete. In [AK91] it was shown that, under some cryptographic assumptions, the problems

*IBM – Thomas J. Watson Research Center. P. O. Box 704, Yorktown Heights, New York 10598. Part of the work was done while the author was at: Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138

of learning DNF with and without membership queries are equivalent. Both results apply only to very specific distributions, and do not seem to extend to the uniform distribution. In [Kha92], the hardness of learning AC^1 circuits, even under uniform distribution, is shown, under specific cryptographic assumption about the hardness of the subset sum.

The work of [LMN89] established the connection between the Fourier spectrum and learnability. They presented a quasi-polynomial-time (i.e. $O(n^{poly-log(n)})$) algorithm for learning the class AC^0 (polynomial size constant depth circuits); the approximation is with respect to the uniform distribution. Their main result is an interesting property of the representation of the Fourier transform of AC^0 circuits; based on it they derived a learning algorithm for AC^0. In [AM91] polynomial time algorithms are given for learning both decision lists and read once decision trees with respect to the uniform distribution. The work of [KM91] uses the Fourier representation to derive a polynomial time learning algorithm for decision trees, with respect to the uniform distribution. The relation between DNFs and their Fourier transform representation is also studied in [BHO90]. Other works that are investigating the Fourier transform of Boolean functions are [Bru90, BS90b, SB91].

The techniques developed in [GL89, KM91] give a randomized polynomial time algorithm that performs the following task. The input is a Boolean function f that can be approximated by a polynomially sparse function g (a function with a polynomial number of non-zero coefficients) such that the expected error square (i.e. $E(f-g)^2$) is bounded by ε. The algorithm finds some polynomially sparse function h that approximates f, such that the expected error square is $O(\varepsilon)$, i.e. $E(f-h)^2 = O(\varepsilon)$.

The main contribution of this work is showing that a DNF with terms of size d can be approximated by a $d^{O(d \log 1/\varepsilon)}$ sparse function, such that the expected error squared is at most ε, with respect to the uniform distribution. This result, in conjunction with the results of [GL89, KM91], gives a learning algorithm that runs in time $d^{O(d \log 1/\varepsilon)}$, and learns a DNF with terms of size d, with respect to the uniform distribution.

When performing an approximation with respect to the uniform distribution we can relate the number of terms and the size of each term. Namely, when considering a DNF with m terms, terms of size larger than $\Omega(\log \frac{m}{\varepsilon})$ may be ignored, since they have a negligible influence on the DNF. This immediately gives an $O(n^{\log m})$ learning algorithm for a DNF with m terms, with respect to the uniform distribution. For a DNF with a polynomial number of terms (i.e. $m = n^{O(1)}$) this immediately gives an $O(n^{\log n})$ learning algorithm. (See [Ver90]).

The results here are mainly interesting in the case that ε is a constant. In this case the algorithm runs in time $O(n^{\log \log n})$ and finds an approximation to a polynomial size DNF, with respect to the uniform distribution. Another consequence of the result is that DNF with terms of size less than $O(\log n / \log \log n)$ can be approximated in time $O(n^{\log 1/\varepsilon})$; again, if ε is constant then the algorithm runs in polynomial time.

An interesting consequence of the algorithm is that it depends on the size of the terms rather than the number of terms. This implies that any DNF with terms of size d can be approximated (for a constant ε) by some $O(d^d)$ sparse function, even if the DNF has $\Omega(n^d)$ terms. This can be contrasted with the results based on the Sunflower Theorem that show that a DNF with terms of size at most d can be approximated by a DNF with $O(2^{d^2})$ terms of size at most d (see [LV91]).

Our results are based on the lower bound techniques that were developed for proving lower bound for polynomial size constant depth circuit [Ajt83, FSS84, Yao85, Has86]. Those techniques work almost identically for DNF and CNF; for this reason all our results apply also to CNF.

The paper is organized as following. Section 2 gives the notation and definition that is used later. Section 3 includes the main results of this work and proves the main theorem based on two lemmas, which are proven later in Section 4 and Section 5. In Appendix A, we analyze a read-once DNF and show that some of the properties that we derive for a general DNF are tight even for a read once DNF.

2 NOTATION

2.1 FOURIER TRANSFORM

Boolean functions on n variables are considered as real valued functions $f : \{0,1\}^n \rightarrow \{-1,1\}$. The set of all real functions on the cube is a 2^n-dimensional real vector space with an inner product defined by:

$$< g, f >= 2^{-n} \sum_{x \in \{0,1\}^n} f(x)g(x) = E(gf)$$

(where E is expectation) and as usual the *norm* of a function is defined: $\|f\| = \sqrt{< f, f >}$, which is the Euclidean norm.

The basis of the cube $\mathbf{Z_2^n}$ is defined as follows: for each subset S of $\{1, \cdots, n\}$, define the function χ_S:

$$\chi_S(x_1, \cdots, x_n) = \begin{cases} +1 & \text{if } \sum_{i \in S} x_i \text{ is even} \\ -1 & \text{if } \sum_{i \in S} x_i \text{ is odd} \end{cases}$$

The following properties of these basis functions can be easily verified:

- For every A, B: $\chi_A \chi_B = \chi_{A \Delta B}$, where $A \Delta B$ is the symmetric difference of A and B.
- The family $\{\chi_S\}$ for all $S \subset \{1 \cdots n\}$ forms an orthonormal basis, i.e., if $A \neq B$, then $< \chi_A, \chi_B >= 0$, and for every A, $< \chi_A, \chi_A >= 1$.

Any real valued function on the cube can be uniquely expressed as a linear combination of the basis functions, i.e. $\sum_S c_S \chi_S$, where c_S are real constants. The Fourier transform of a function is writing it as a linear combination of the χ_S's. For a function f and $S \subset \{1, \cdots, n\}$, the S'th Fourier coefficient of S denoted by $\hat{f}(S)$ is what was previously called c_S, i.e., $f = \sum_S \hat{f}(S) \chi_S$.

Since the χ_S's are an orthonormal basis, Fourier coefficients are found via:

$$\hat{f}(S) =< f, \chi_S >$$

For Boolean f this specializes to:

$$\hat{f}(S) = Pr[f(x) = \oplus_{i \in S} x_i] - Pr[f(x) \neq \oplus_{i \in S} x_i]$$

where $x = (x_1, x_2, \ldots, x_n)$ is chosen uniformly at random in $\{0,1\}^n$.

The orthonormality of the basis implies Parseval's identity:

$$\|f\|^2 = \sum_{S \subset \{1 \cdots n\}} \hat{f}(S)^2$$

Note that if f is Boolean then $\|f\| = 1$.

A *t-sparse function* is a function that has at most t non-zero coefficients. The `Fourier degree` of a Boolean function, denoted by $\texttt{F-deg}(f)$, is the size of the largest set S such that $\hat{f}(S) \neq 0$. Note that this equals the degree of f as a real (multi-linear) polynomial.

2.2 RANDOM RESTRICTION

The technique of random restriction was introduced in [FSS84] in order to derive lower bounds for AC^0 circuits. It was latter used in [Yao85, Has86] to improve the lower bounds. (See [BS90a] for an excellent survey on the subject.)

A restriction ρ is a mapping of the input variables to 0, 1 and $*$. The function obtained from $f(x_1, \cdots, x_n)$ by applying a restriction ρ is denoted by f_ρ. The inputs of f_ρ are those x_i for which $\rho(x_i) = *$, while all other variables are set according to ρ.

The set of *live variables* with respect to a restriction ρ is the set of variables that is assigned the value $*$, this set is denoted by $live(\rho) = \{x_i | \rho(x_i) = *\}$. A *random restriction ρ with a parameter p* is obtained by setting each x_i, independently, to a value from $\{*, 0, 1\}$, such that $Pr[\rho(x_i) = *] = p$, and $Pr[\rho(x_i) = 1] = Pr[\rho(x_i) = 0] = \frac{1-p}{2}$.

2.3 LEARNING MODEL

The learning model here uses membership queries and the learning is done with respect to the uniform distribution. We assume that the learning algorithm is given some unknown function which it can access only as a black box, i.e., it can query the unknown function f, on any input $x \in \{0,1\}^n$, and receive $f(x)$.

The learning algorithm, after performing a finite number of membership queries, outputs a hypothesis h. The error of a hypothesis h, with respect to the function f, is defined to be $error(f, h) \stackrel{\triangle}{=} Prob[f(x) \neq h(x)]$, where x is distributed uniformly over $\{0,1\}^n$.

An algorithm A *learns* a class of functions $\mathcal{F}$, if for every $f \in \mathcal{F}$, and $\varepsilon, \delta > 0$, the algorithm outputs an hypothesis $h = A(f, \varepsilon, \delta)$, such that

$$Prob[error(f, h) \geq \varepsilon] \;\leq\; \delta.$$

2.4 BOOLEAN PREDICTION

A (real valued) function g *ε-approximates* a function f in norm L_2 if $E[(f-g)^2] \leq \varepsilon$. In the case that f is a Boolean function, we can convert a prediction function g to a Boolean prediction by predicting the sign of g. More formally, for a real number r let $sign(r) = +1$ if r is positive and $sign(r) = -1$ otherwise. Note that if $f(x) \neq sign(g(x))$ then $|f - g| \geq 1$, which implies,

$$Prob[f \neq sign(g)] \leq E[(f-g)^2] \leq \varepsilon.$$

Thus, we have

Claim 2.1 *If g ε-approximates a Boolean function f in L_2 norm, then*

$$Prob[f \neq sign(g)] \leq \varepsilon.$$

2.5 DECISION TREES

A *decision tree* consists of a labeled binary tree. Each inner node v of the tree is labeled by an input x_v and has two outgoing edges and each leaf of the tree is labeled by either $+1$ or -1.

An input to the decision tree defines a *computation.* The computation starts at the root and traverses a path to a leaf. When the computation arrives at an inner node v, labeled by input x_v, if $x_v = 1$ the computation continues to the right son of v, otherwise it continues to the left son. The computation terminates at a leaf u and outputs the label of u.

The depth of a decision tree T, denoted by DT-depth(T), is the maximum path in T. For a function f we denote by DT-depth(f) the minimum depth of a decision tree that computes f.

3 THE MAIN RESULTS

The main contribution of this paper is to show that any DNF with terms of size d can be ε-approximated by an M-sparse function, where $M \leq d^{O(d \log \frac{1}{\varepsilon})}$. This result translates into a learning algorithm, under the uniform distribution, for DNF with terms of size d, and the running time of the algorithm is polynomial in M. Our main theorem states the following.

Theorem 3.1 *For any function f that can be described by a DNF with terms of size d there exists an M-sparse function g that ε-approximates f and $M \leq d^{O(d \log \frac{1}{\varepsilon})}$.*

The proof of the above theorem is based on combining two Lemmas. The first Lemma shows properties of the coefficients of "large" sets, while the second deals with properties of "small" sets. (Sets of size $\Omega(d \log 1/\varepsilon)$ are considered "large" sets while sets of size $O(d \log 1/\varepsilon)$ are considered "small".)

The following lemma shows that the sum of the squares of all the coefficients that corre-sponding to "large" sets is negligible.

Lemma 3.2 *Let f be a function that can be written as a DNF with terms of size d. Then,*

$$\sum_{|S| > 20d \log \frac{4}{\varepsilon}} \hat{f}^2(S) \leq \frac{\varepsilon}{2}$$

The proof of the above lemma is found is Section 4, and it is essentially a special case of the proof in [LMN89].

In Lemma 3.3, which is proved in Section 5, we restrict our attention to coefficients of sets of size at most τ. We show that the sum of the absolute values of the coefficients of all the sets of size at most τ is bounded by $d^{O(\tau)}$.

Lemma 3.3 *If a function f can be described by a DNF with terms of size d then*

$$\sum_{S:|S|\leq\tau} |\hat{f}(S)| \leq 4(20d)^{\tau} = d^{O(\tau)}$$

Based on the above two lemmas (which are proven latter) we prove the main theorem.

Proof of Theorem 3.1: Given a function f, that is described by a DNF with terms of size d, we need to exhibit a function g that ε-approximates f. The function g that we exhibits has `Fourier degree` of $\tau = 20d\log\frac{4}{\varepsilon}$, i.e., $\hat{g}(S) = 0$, for $|S| > \tau$. Lemma 3.2 shows that forcing the "large" coefficients to zero might increase the expected error square by at most $\varepsilon/2$.

Lemma 3.3 gives a property of sets of size less than τ. This property shows that the sum, in absolute value, of the coefficients of the "small" sets, is small. Namely, let $L = \sum_{|S|\leq\tau} |\hat{f}(S)|$, then $L \leq d^{O(\tau)}$. In a similar way to [KM91] we show that if the sum in absolute value of the coefficients is L, then there can be at most $(\frac{2L}{\varepsilon})^2$ "interesting" coefficients. The definition of an "interesting" coefficient of f is a coefficient that in absolute value is more than $\varepsilon/2L$, and which correspond to a set of size at most τ. The function g includes all those coefficients of f, and no other coefficient.

More formally, let $G = \{S \ : \ |\hat{f}(S)| \geq \varepsilon/2L \text{ and } |S| \leq \tau\}$ and $g(x) = \sum_{S\in G} \hat{f}(S)\chi_S(x)$, where $L = \sum_{|S|\leq\tau} |\hat{f}(S)|$. Since $|G| \leq (2L/\varepsilon)^2$, the function g is M-sparse, where $M \leq (2L/\varepsilon)^2$. By Lemma 3.3, $L = d^{O(\tau)}$, and therefore M has the bound claimed in the statement of the theorem.

By the definition of g, and using the Parseval identity, we have

$$\begin{aligned} E[(f-g)^2] &= \sum_{S\notin G} \hat{f}^2(S) \\ &= \sum_{|S|>\tau} \hat{f}^2(S) + \sum_{|S|\leq\tau \text{ and } |\hat{f}(S)|<\varepsilon/2L} \hat{f}^2(S) \end{aligned}$$

By Lemma 3.2, the first summation is bounded by $\varepsilon/2$. We bound the second summation as follows.

$$\begin{aligned} \sum_{|S|\leq\tau \text{ and } |\hat{f}(S)|<\varepsilon/2L} \hat{f}^2(S) &\leq \left(\max_{S\notin G \text{ and } |S|\leq\tau} |\hat{f}(S)|\right) \cdot \left(\sum_{|S|\leq\tau} |\hat{f}(S)|\right) \\ &\leq \frac{\varepsilon}{2L} \cdot L = \varepsilon/2. \end{aligned}$$

This implies that $E[(f-g)^2] \leq \varepsilon$, which implies that g ε-approximates f. $\Box$

The main application of theorem 3.1 is to exhibit various learning algorithms for DNF. The learning algorithms are based on the following result of [KM91].

Theorem 3.4 ([KM91]) *Let f be a Boolean function such that there exists a t-sparse function g that ε-approximates f. There exists a randomized algorithm, whose running time is polynomial in t, n, $1/\varepsilon$ and $\log \frac{1}{\delta}$, that given f (as a black box) and $\delta > 0$ outputs a function h, such that the probability that h $O(\varepsilon)$-approximates f is at least $1-\delta$.*

The above theorem implies that in our case the running time of the learning algorithm would be polynomial in $d^{O(d \log \frac{1}{\varepsilon})}$, n and $\log 1/\delta$.

Corollary 3.5 *There exists an algorithm that given (as a black box) a function f, that can be described by a DNF with terms of size d, computes an M-sparse function g, where $M \leq 64(20d)^{2\tau}/\varepsilon^2$, and $\tau = 20d \log \frac{4}{\varepsilon}$, such that the probability that g ε-approximates f is at least $1-\delta$. The running time of the algorithm is bounded by a polynomial in $d^{d \log 1/\varepsilon}$, n and $\log 1/\delta$.*

When considering learning a DNF with m terms, with respect to the uniform distribution, terms of size larger then $\Omega(\log(m/\varepsilon))$ may be ignored (this may increase the expected error squared by at most $O(\varepsilon)$). This implies that when considering a DNF with a polynomial number of terms, we can set $d = O(\log \frac{n}{\varepsilon})$ and derive the following corollary.

Corollary 3.6 *There exists an algorithm that learns, with respect to the uniform distribution, the class of DNF with a polynomial number of terms, and the running time of the algorithm is bounded by a polynomial in $(\frac{n}{\varepsilon})^{\log\log \frac{n}{\varepsilon} \log \frac{1}{\varepsilon}}$ and $\log \frac{1}{\delta}$.*

Note that Corollary 3.6 gives an $n^{O(\log\log n)}$ running time for a constant ε. Another interesting case is bounding the size of the terms of the DNF to be at most $O(\log n / \log\log n)$. For this case we have the following corollary.

Corollary 3.7 *There exists an algorithm that learns, with respect to the uniform distribution, the class of DNF terms of size $O(\log n/\log\log n)$, and the running time of the algorithm is bounded by a polynomial in $n^{\log 1/\varepsilon}$ and $\log 1/\delta$.*

Again, in the case that ε is a constant the running time in Corollary 3.7 is polynomial in n.

4 THE FOURIER DEGREE BOUND

In this section we consider a property of the "large" coefficients of a DNF with terms of size d. We show that the sum of squares of the coefficients that correspond to sets larger than a certain threshold $\tau(d)$ is bounded by ε. This intuitively implies that when approximating such a DNF, we can ignore the coefficients of sets larger than $\tau(d)$. The main goal of this section is proving Lemma 3.2, which was already claimed and used in Section 3.

Implicit in the proofs of [LMN89] is the following connection between a random restriction of a function and the coefficients of "large" sets of the function.

Lemma 4.1 ([LMN89]) *For any Boolean function f,*

$$\sum_{|S|>t} \hat{f}^2(S) \leq 2Prob_\rho[\texttt{F-deg}(f_\rho) \geq tp/2]$$

Note that the result is non-trivial only if $Prob_\rho[\texttt{F-deg}(f_\rho) \geq tp/2] \leq \frac{1}{2}$. As in [LMN89], we use the Switching lemma, from [Has86], stated below.

Lemma 4.2 (Hastad) *Let f be given by a DNF formula where each term has size at most d, and choose a random restriction ρ with parameter p (i.e. $Pr[\rho(x_i) = *] = p$). Then*

$$Prob_\rho[\texttt{DT-depth}(f_\rho) \geq s] \leq (5pd)^s$$

Consider a decision tree T of depth s and a set U of variables of size at least $s+1$. The path to each leaf includes at most s variables, therefore, when considering the parity on the subset U of the variables, half of the inputs that reach the leaf have parity one, while the other half parity zero. Therefore the Fourier coefficients of T, for any set larger than s, are zero (see [LMN89]). This implies that $\texttt{F-deg}(f_\rho) \leq \texttt{DT-depth}(f_\rho)$, which immediately bounds $Prob_\rho[\texttt{F-deg}(f_\rho) \geq tp/2]$ from above by $Prob_\rho[\texttt{DT-depth}(f_\rho) \geq tp/2]$. Combining this with Lemma 4.1, Lemma 4.2 and setting $p = 1/(10d)$ and $t = 20d \log \frac{4}{\varepsilon}$ proves Lemma 3.2.

Lemma 3.2 *Let f be a function that can be written by a DNF with terms of size d. Then,*

$$\sum_{|S|>20d\log\frac{2}{\varepsilon}} \hat{f}^2(S) \leq \frac{\varepsilon}{2}$$

The above Lemma demonstrates that in order to approximate a DNF with terms of size d, it is sufficient to consider the coefficients of sets of size at most $\tau = O(d \log \frac{1}{\varepsilon})$. The problem is that there is a large number of such coefficients. In the next section we show that only a "few" of those coefficients are "large".

It is interesting to note that the bound for the `Fourier degree` in Lemma 3.2 is tight in the following sense.

Theorem 4.3 *There exists a function, f, that can be described by a DNF with terms of size d, such that no function with* `Fourier degree` *less than $\frac{1}{2}d\log\frac{1}{\varepsilon}$ can ε approximate f.*

Proof: To prove the theorem it is sufficient to show a function, f, that can be described by a DNF with terms of size d, and for some subset S, $|S| = \frac{1}{2}d\log\frac{1}{\varepsilon}$, has the property that $|\hat{f}(S)| \geq \sqrt{\varepsilon}$. The construction of such a function appears, essentially, in [Has86] and is described below for completeness.

Consider the following function. Of the n inputs, the function uses only $\frac{1}{2}d\log 1/\varepsilon$ variables as its support. The function has $1/2\log 1/\varepsilon$ disjoint blocks of input variables, each block of size d. The function computes the parity in each block and outputs the OR of the parities. The function can be computed by a DNF with $2^{d-1}\log 1/\varepsilon$ terms of size d.

The OR function of k variables has $\hat{OR}(T) = 2\cdot 2^{-k}$, for $T \neq \emptyset$; in particular this holds for $T = \{1\cdots k\}$.

Consider the set S that includes all the variables in the support of the function. We show that the coefficient of the set S is $2\sqrt{\varepsilon}$. Note that in our construction the parity of the inputs in S equals the parity of the outputs of the blocks (which is the parity within each block). Therefore the coefficient $\hat{f}(S) = 2\cdot 2^{-1/2\log 1/\varepsilon} = 2\sqrt{\varepsilon}$, which concludes the proof of the theorem. $\square$

In the appendix we show that even a read-once DNF, with terms of size d requires `Fourier degree` of at least $\Omega(d\log\frac{1}{\varepsilon}/\log\log\frac{1}{\varepsilon})$ for its approximation.

5 BOUNDING COEFFICIENTS OF SMALL SETS

In the previous section we bounded the magnitude of the coefficients of large sets. In this section we focus of coefficients of small sets. While any specific coefficient of a set of size less than $\tau = 20d\log 4/\varepsilon$ can potentially be "significant", in order to achieve a good approximation, we show that only a relatively small number of such coefficients can be simultaneously "significant". This is done by bounding the sum in absolute value of those coefficients. In the derivation of the bounds we use the following definitions.

Definition 1 *Let*

$$L_{1,k}(f) = \sum_{|S|=k} |\hat{f}(S)|,$$

and

$$L_1(f) = \sum_{i=0}^{n} L_{1,k}(f) = \sum_{S} |\hat{f}(S)|$$

Our main aim is to bound $L_{1,k}(f)$ by $d^{O(k)}$, where d is the size of the largest term in a DNF representation of f.

The proof uses the fact that after a random restriction, the restricted DNF can be written as a decision tree with small depth. Recall that the depth of a decision tree is the longest path from the root to a leaf and is denoted by $\texttt{DT-depth}(T)$.

Another tool that we use here is that we can bound the L_1 norm of the coefficients of a decision tree. The following is a special case of the bound in [KM91].

Lemma 5.1 ([KM91]) *For a function f, if* $\texttt{DT-depth}(f) \leq s$ *then* $L_1(f) \leq 2^s$.

We start by showing that after a random restriction the L_1 norm of the restricted function is very small.

Lemma 5.2 *Let f be given by a DNF formula where each term has size at most d. Let ρ be random restriction with parameter $p \leq \frac{1}{20d}$, then*

$$E_\rho[L_1(f_\rho)] \leq 2$$

Proof: We can express the expectation as,

$$E_\rho[L_1(f)] = \sum_{s=0}^{n} Prob[\texttt{DT-depth}(f_\rho) = s] \cdot E_\rho[L_1(f_\rho) \mid \texttt{DT-depth}(f_\rho) = s]$$

By Lemma 5.1, for any ρ, such that $\texttt{DT-depth}(f_\rho) = s$, $L_1(f_\rho) \leq 2^s$. By Lemma 4.2

$$Prob[\texttt{DT-depth}(f_\rho) \geq s] \leq (5dp)^s$$

Therefore,

$$E_\rho[L_1(f)] \leq \sum_{s=0}^{n} (5pd)^s 2^s = \sum_{s=0}^{n} (10pd)^s.$$

For $p \leq \frac{1}{20d}$, the sum is less than two, and the lemma follows. □

The next lemma establishes the connection between $L_{1,k}(f)$ and the value of $E_\rho[L_{1,k}(f_\rho)]$.

Lemma 5.3 *Let f be a Boolean function and ρ a random restriction with parameter p (i.e., $Prob[x_i = *] = p$), then*

$$L_{1,k}(f) \leq (\frac{1}{p})^k E_\rho[L_{1,k}(f_\rho)]$$

Proof: Consider a random variable $\mathcal{L} \subset \{x_1 \cdots x_n\}$, such that for each x_i, independently, $Prob[x_i \in \mathcal{L}] = p$. The random variable $\mathcal{L}$ is the set of live variables in a random restriction with parameter p. We can rewrite $L_{1,k}$ in the following way.

$$L_{1,k}(f) = \sum_{|S|=k} |\hat{f}(S)| = (\frac{1}{p})^k E_{\mathcal{L}}[\sum_{S \subset \mathcal{L} \& |S|=k} |\hat{f}(S)|].$$

Note that in both summations we are summing the original coefficients of the function.

Consider an arbitrary choice for $\mathcal{L}$ and a subset $S \subset \mathcal{L}$.

$$\begin{aligned} |\hat{f}(S)| &= |E_{x_1,\ldots,x_n}[f(x_1,\ldots,x_n)\chi_S(x_1,\ldots,x_n)]| \\ &\leq E_{x_i \notin \mathcal{L}} |E_{x_j \in \mathcal{L}}[f(x_1,\ldots,x_n)\chi_S(x_1,\ldots,x_n)]| \\ &= E_\rho[|\hat{f}_\rho(S)| \mid live(\rho) = \mathcal{L}] \end{aligned}$$

The last line follows from the observation that averaging over $x_i \notin \mathcal{L}$ is the same as taking the expectation of a random restriction whose set of live variables is restricted to be $\mathcal{L}$.

Since the absolute value of every coefficient S is expected to increase, this implies that

$$\sum_{S \subset \mathcal{L} \& |S|=k} |\hat{f}(S)| \leq E_\rho[\sum_{S \subset \mathcal{L} \& |S|=k} |\hat{f}_\rho(S)| \mid live(\rho) = \mathcal{L}] = E_\rho[L_{1,k}(f_\rho) \mid live(\rho) = \mathcal{L}]$$

Now we can go back and use the first equality we derived. In that equality we are averaging over $\mathcal{L}$. Therefore,

$$L_{1,k}(f) = (\frac{1}{p})^k E_{\mathcal{L}}[\sum_{S \subset \mathcal{L} \& |S|=k} |\hat{f}(S)|] \leq (\frac{1}{p})^k E_\rho[L_{1,k}(f_\rho)],$$

which completes the proof of the lemma. □

We can now prove Lemma 3.3, the second lemma used in Section 3.

Lemma 3.3 *If a function f can be described by a DNF with terms of size d,then*

$$\sum_{S:|S|\leq\tau} |\hat{f}(S)| \leq 4(20d)^\tau = d^{O(\tau)}$$

Proof: Note that $\sum_{S:|S|\leq\tau} |\hat{f}(S)| = \sum_{k=0}^{\tau} L_{1,k}(f)$. By setting $p = \frac{1}{20d}$, and combining Lemma 5.2 and Lemma 5.3, we have that

$$L_{1,k}(f) \leq 2(20d)^k$$

and the lemma follows. □

In the Appendix we show that the above bound is almost tight even for a read-once DNF f. We show that for k between $\sqrt{d}$ and $2^{\sqrt{d}}$, the value of $L_{1,k}(f)$ is at least $d^{k/4}$.

Acknowledgements

I would like to thank Eyal Kushilevitz for commenting on an early version of the paper.

References

[Ajt83] M. Ajtai. $\sum_1^1$-formulae on finite structure. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[AK91] Dana Angluin and Michael Kharitonov. When won't membership quiries help? In *Proceedings of STOC '91*, pages 44–454. ACM, 1991.

[AM91] William Aiello and Milena Mihail. Learning the fourier spectrum of probabilistic lists and trees. In *Proceedings SODA 91.* ACM, Jan 1991.

[AP91] Howard Aizenstein and Leonard Pitt. Exact learning of read-twice DNF formulas. In *32nd Annual Symposium on Foundation of Computer Science*, pages 170–179, October 1991.

[BHO90] Y. Brandman, J. Hennessy, and A. Orlitsky. A spectral lower bound technique for the size of decision trees and two level circuits. *IEEE Trans. on Computers.*, 39(2):282–287, 1990.

[Bru90] J. Bruck. Harmonic analysis of polynomial threshold functions. *Siam J. on Disc. Math.*, 3(2):168–177, May 1990.

[BS90a] R. Boppana and M. Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of theor. comp. sci.* 1990.

[BS90b] J. Bruck and R. Smolensky. Polynomial threshold functions, AC^0 functions and spectral norms. In 31^{th} *Annual Symposium on Foundations of Computer Science, St. Louis, Missouri*, pages 632–641, October 1990.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proc. 21st ACM Symposium on Theory of Computing*, pages 25–32. ACM, 1989.

[Han91] Thomas Hancock. Learning 2μ DNF and $k\mu$ decision trees. In *4th COLT*, pages 199–209, August 1991.

[Has86] J. Hastad. *Computational limitations for small depth circuits.* MIT Press, 1986. Ph.D. thesis.

[HM91] Thomas Hancock and Yishay Mansour. Learning monotone $k\mu$ DNF formulas on product distributions. In *COLT*, pages 179–183, August 1991.

[Kha92] Michael Kharitonov. When won't membership quiries help? In *Proceedings of COLT '92*, pages 29–36. ACM, 1992.

[KLPV87] Michael Kearns, Ming Li, Leonard Pitt, and Leslie Valiant. On the learnability of Boolean formulae. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 285–295, New York, New York, May 1987.

[KM91] E. Kushilevitz and Y. Mansour. Learning decision trees using the fourier spectrum. In *Proceedings of the* 23^{rd} *Annual ACM Symposium on Theory of Computing*, May 1991. to appear.

[LMN89] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform and learnability. In 30^{th} *Annual Symposium on Foundations of Computer Science, Reseach Triangle Park, NC*, pages 574–579, October 1989.

[LV91] M. Luby and B. Velickovic. On deterministic approximation of DNF. In *Proceedings of the* 23^{rd} *Annual ACM Symposium on Theory of Computing*, pages 430–438, May 1991.

[PV88] Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988.

[SB91] Kai-Yeung Sui and Jehoshua Bruck. On the power of threshold circuits with small weights. *Siam J. on Disc. Math.*, 4(3):423–435, Aug 1991.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[Ver90] Karsten Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of COLT '90*, pages 314–326, 1990.

[Yao85] A. C. Yao. Separating the polynomial-time hierarchy by oracles. In 26^{th} *Annual Symposium on Foundations of Computer Science, Portland, Oregon*, pages 1–10, October 1985.

A READ-ONCE DNF

In this section, we analyze the Fourier transform of a read-once DNF. We show that the bound that we proved for the `Fourier degree` and for $L_{1,k}$ are almost tight, even for read once DNFs. We also prove a lower bound on the sparsity required by any function approximating a read once DNF.

The DNF that we analyze has m terms each of size d, and the terms are disjoint. First, we compute the Fourier coefficients of this DNF and later we show the properties based on this analysis. We start by computing the Fourier transform of the AND function.

The AND function is define as follows. $AND(x_1, \ldots, x_d) = 1$ iff each input x_i equals 1, and otherwise $AND(x_1, \ldots, x_d) = 0$. (The range of the AND is $\{0, 1\}$; this fact simplifies the next step in the construction.) The Fourier transform of the AND function is,

$$AND(x_1, \ldots, x_d) = \prod_{i=1}^{d} \frac{1 - \chi_{\{i\}}(x)}{2} = \frac{1}{2^d} \sum_{S \subset \{1,\ldots,d\}} (-1)^{|S|} \chi_S(x)$$

This means that the S'th Fourier coefficient of the function AND is $\hat{AND}(S) = (-1)^{|S|} 2^{-d}$. (We can also use this to observe that $L_1(AND) = 1$.)

The next step is to compute the OR of m different AND functions (i.e., terms). We define $OR(y_1, \ldots, y_m) = +1$ iff all the y_i are equal to 0, and otherwise $OR(y_1, \ldots, y_m) = -1$. We can write the OR function as,

$$OR(y_1, \ldots, y_m) = 2 \left(\prod_{i=1}^{m} (1 - y_i) \right) - 1$$

In constructing a read-once DNFs we let $y_i = AND(x_{i,1}, \ldots, x_{i,d})$.

Since the different ANDs have different input variables, we can divide each subset S of the dm variables to m disjoint subsets, $S_1, \cdots S_m$, where $S_i \subset \{x_{i,1}, \ldots, x_{i,d}\}$. Let f be a read-once DNF with m terms of size d. The Fourier coefficients of f are,

$$\hat{f}(\emptyset) = 2(1 - 2^{-d})^m - 1$$

and

$$\hat{f}(S_1, \ldots, S_m) = 2(-1)^k (-1)^{\sum |S_i|} 2^{-dk} (1 - 2^{-d})^{m-k}$$

where $k \geq 1$ is the number of S_i such that $S_i \neq \emptyset$. The absolute value of a coefficient can be written as,

$$|\hat{f}(S_1, \ldots, S_m)| = 2(1 - 2^{-d})^m (2^d - 1)^{-k}$$

where $k \geq 1$ is the number of S_i such that $S_i \neq \emptyset$. (Note that $L_1(f) \geq 2^m (1 - 2^{-d})^m$.)

From now on we assume that $m = 2^d$, which implies that the probability that $f = 0$ is $1/e$. The first theorem shows that the degree bound is almost tight even for a read once DNF.

Theorem A.1 *Let f be a read-once DNF with 2^d terms each of size d. The function f cannot be ε-approximated by a function with* `Fourier degree` *less than $\frac{1}{2} d \log \frac{1}{\varepsilon} / \log \log \frac{1}{\varepsilon}$.*

Proof: A function with `Fourier degree` less than τ has all the coefficients of sets of size τ identically zero. A lower bound on the expected error squares of any such function and f is the sum of square of the coefficients of sets of size τ of f. Therefore if we can show that this sum of squares is at least ε then the expected error squared is at least ε.

Consider all the coefficients of f that have exactly k non-empty sets S_i, and each non-empty set has at least $d/2$ elements (i.e., $|S_i| \geq d/2$). There are

$$\binom{m}{k} 2^{(d-1)k}$$

such sets. We are interested in the sum of the squares of the coefficients of those sets. Each coefficient squared equals $4(1-2^{-d})^{2m}(2^d-1)^{-2k}$. Therefore the sum equals,

$$\binom{m}{k} 2^{(d-1)k} 4(1-2^{-d})^{2m}(2^d-1)^{-2k}.$$

For $m = 2^d$, the sum is bounded from below by

$$(\frac{2^d}{k})^k 2^{(d-1)k} 4e^{-2} 2^{-2dk} = 4e^{-2}(2k)^{-k}$$

This expression is smaller than ε only for $k \geq \frac{1}{2}\log\frac{1}{\varepsilon}/\log\log\frac{1}{\varepsilon}$. Since each is a coefficient of a set with exactly $kd/2$ inputs, the theorem follows. $\square$

The above proof shows something even stronger than a bound on the degree of the approximating function. It shows also a bound on the sparsity of the approximating function. Since the approximating function needs to include most of the coefficients that were mentioned in the proof, we can establish the following result about the sparsity of the approximating function.

Theorem A.2 *Let f be a read-once DNF with $m = \Theta(2^d)$ terms each of size d and g an M-sparse function. If g ε-approximates f then $M = \Omega((\frac{m2^d}{2\tau})^\tau)$, where $\tau = \Omega(\log\frac{1}{\varepsilon}/\log\log\frac{1}{\varepsilon})$.*

The third theorem shows that the bound that we derived for $L_{1,k}(f)$ is almost tight.

Theorem A.3 *Let f be a read-once DNF with with 2^d terms each of size d. Then $L_{1,k}(f) \geq d^{k/4}$, for $\sqrt{d} \leq k \leq 2^{\sqrt{d}}$.*

Proof: Consider the function f, and for a given k consider the coefficients that involve $l = \lfloor k/\sqrt{d} \rfloor$ non-empty sets, where each non-empty set has exactly $\sqrt{d}$ elements. The number of such sets is,

$$\binom{m}{l}\binom{d}{\sqrt{d}}^l$$

Each coefficient in absolute value is

$$2(1-2^{-d})^m(2^d-1)^{-l} \geq 2^{-dl-3},$$

where the bound uses the fact that $m = 2^d$. Therefore,

$$L_{1,k}(f) \geq (m/l)^l d^{l\sqrt{d}/2} 2^{-dl-3} = d^{k/2}/(8l^l)$$

For $k \leq 2^{\sqrt{d}}$ and $d^{k/4} \geq 2l^l$ the theorem follows. $\square$