To appear in Information Processing Letters

On Using the Fourier Transform to Learn Disjoint DNF

Roni Khardon*

Aiken Computation Laboratory
Harvard University

November 29, 1993

Keywords: Algorithms,Computational Learning,Membership Queries, DNF,Fourier Transform.

1 Introduction

This note addresses the problem of learning DNF expressions using membership queries. We extend the results of Kushilevitz and Mansour [5], on learning through Fourier representations, to show learnability of various subsets of DNF. In particular we show the polynomial learnability of Disjoint DNF expressions under the uniform distribution, and exact learnability of Disjoint $\log n$ DNF. These extend the learnability results of the corresponding classes of decision trees given in [5]. We further show the learnability of $\log n$ term DNF under the uniform distribution. The learnability of this class (even for the distribution free case) is already known [2], but the algorithm and analysis given here are different. The learning framework and algorithm are the same as in [5]. The main contribution of this note is a different analysis of the Fourier spectrum of these function classes. This enables us to show the learnability of wider classes and with somewhat simplified proofs.

The learning results are in the framework of PAC learnability, and exact learnability, with membership queries (no random examples are used). Namely, the learning algorithm uses membership queries and with high probability outputs a "good" hypothesis. In the PAC model, a good hypothesis has a small probability of error when used to predict the value of the target function. In the exact learnability model the hypothesis is not allowed to make an error on any input.

Disjoint DNF expressions are DNF expressions in which every truth assignment satisfies at most one term. The learnability of this class has been studied before [1, 4], and it strictly includes the class of decision trees. An example in [4] shows that disjoint DNF strictly includes the class of DNF intersection CNF (which includes decision trees), so that our results are not implied by the recent learnability result for the latter [3].

Fourier Transform and Learnability

Every function over the boolean cube $\{0,1\}^n$ can be viewed as a vector with 2^n entries. Each entry is the value of the function on the input that corresponds to that entry. This description allows us to consider function classes as subspaces of the vector space with 2^n entries (and arbitrary real values). The dimension of this vector space is clearly 2^n.

*e-mail: roni@das.harvard.edu . Research supported by the U.S. Army Research Office under grant DAAL03-92-G-0164 (Center for Intelligent Control Systems).

The base of characters (known as the Fourier base) have been shown useful for learnability. Each function in this base is an xor operation on some subset of the input bits, and the function takes values in $\{-1,1\}$. Formally, let $Z \subseteq \{1,\ldots,n\}$, then

$$\chi_Z(y_1,\ldots,y_n) = \Pi_{i\in Z}(-1)^{y_i}$$

When using the inner product $< g,f >= 2^{-n}\sum_{x\in\{0,1\}^n} g(x)f(x)$, the Fourier base is orthonormal. That is $\forall Z,S \subseteq \{1,\ldots,n\}$, $Z \neq S$ implies $< \chi_Z,\chi_S >= 0$ and $\forall Z,\ < \chi_Z,\chi_Z >= 1$. This implies the standard representation of functions in an orthonormal base, namely for any real function on the boolean cube $f(y) = \sum_{Z\subseteq\{1,\ldots,n\}} \hat{f}(Z)\chi_Z(y)$, where $\hat{f}(Z) =< f,\chi_Z >$. In the proofs we would implicitly use the following additive property: if $h = f + g$ then $\hat{h}(Z) = \hat{f}(Z) + \hat{g}(Z)$ (this is simply implied by the linearity of the inner product).

In the rest of the paper we assume that the domain of the learned functions is $\{0,1\}^n$, and the learning algorithm is allowed time polynomial in n and in the size of the representation for the target function. The following notation will be used. Let D be a probability distribution on $\{0,1\}^n$, and let U be the uniform distribution on that domain. Let $E_D(\alpha)$ denote the expectation of the random variable α with respect to the distribution D, and let $Prob_D[\beta]$ denote the probability that the 0-1 variable β equals 1 with respect to D. A function f is $t-sparse$ if its representation in the Fourier base requires no more than t non-zero coefficients. The next list (of 3 items) describes some of the results of Kushilevitz and Mansour [5] that we use to derive our result.

1. The first result is $Prob_D[f \neq \text{sign}(g)] \leq E_D[(f-g)^2]$. This implies that it is enough to get a good "squared error" approximation to a function f with respect to D in order to have a good prediction algorithm for it (with respect to the same distribution).

2. If f can be approximated with squared error with respect to U by a function g that is $t-sparse$, then it can be approximated by taking the big coefficients of f and setting other coefficients to zero. Formally, it can be approximated by h such that $\hat{f}(S) \leq \epsilon/t \Rightarrow \hat{h}(S) = 0$ and $\hat{f}(S) > \epsilon/t \Rightarrow \hat{h}(S) = \hat{f}(S)$.

3. There is an algorithm that uses membership queries and finds all the "big" coefficients of a function f for the Fourier base. More formally, with high probability the algorithm finds all coefficients such that $|\hat{f}(S)| > \theta$ and no coefficient such that $|\hat{f}(S)| < \theta/2$. The algorithm runs in time polynomial in $n, 1/\theta$ and $\log 1/\delta$, where θ and δ are its inputs denoting the size of coefficients it looks for and the failure probability it is allowed.

Definition 1 A function class F is learnable with membership queries with respect to distribution *D, if there exists an algorithm A such that $\forall f \in F$, when given access to a membership oracle for f and on input ϵ, δ, the algorithm A runs in polynomial time and with probability at least $1-\delta$ outputs a function h such that $Prob_D[f(x) \neq h(x)] < \epsilon$.*

Definition 2 A function class F is exactly learnable with membership queries*, if there exists an algorithm A such that $\forall f \in F$, when given access to a membership oracle for f and on input δ, the algorithm A runs in polynomial time and with probability at least $1-\delta$ outputs a function h such that $\forall x,\ f(x) = h(x)$.*

The following theorem is implied by the 3 items on the list above. A result similar to Lemma 1 is implicit in the proof of item 2.

Theorem 1 ([5]) *If* $\forall f \in F\ \exists g$ *such that* $E_U[(g-f)^2] < \epsilon$ *and* g *is* $t-sparse$*, then* F *is learnable using membership queries, with respect to the uniform distribution, in time (and queries) complexity polynomial in* $n, t, 1/\epsilon$ *and* $\log 1/\delta$*.*

Lemma 1 *Any class of* $t-sparse$ *functions is exactly learnable with membership queries in time (and queries) complexity polynomial in* n, t *and* $\log 1/\delta$*.*

Proof: Let f be a $t - sparse$ function, and let $T = \{S \mid \hat{f}(S) \neq 0\}$, it is clear that $|T| \leq t$. Fix $\epsilon = 1/4$, we use the algorithm to find all coefficients bigger than ϵ/t, and then we approximate each coefficient to within ϵ/t to create h. Our hypothesis will be sign(h). By the performance guarantee of the algorithm (with high probability) it would not find any coefficient not in T, and therefore we get that $\forall x$,

$$|f(x) - h(x)| = |\sum_{S\in T}(\hat{f}(S) - \hat{h}(S))\chi_S(x)| \leq \sum_{S\in T}|\hat{f}(S) - \hat{h}(S)| \cdot |\chi_S(x)| \leq t \cdot (\epsilon/t) = \epsilon < 1/2$$

This implies $\forall x,\ \ f(x) = \text{sign}(h(x))$. ■

2 The Learning Results

In analyzing the learnability of DNF we would use a characterization of the coefficients of conjunctions, described in the next lemma.

Lemma 2 *Let* $t = l_1 l_2 \ldots l_k$ *be a conjunction (of* k *literals). Then* t *has exactly* 2^k *coefficients not equal to zero in the Fourier base; furthermore* $\hat{t}(Z) \neq 0$ *implies* $|\hat{t}(Z)| = 2^{-k}$*.*

Proof: Let t be the conjunction, and let Z be a subset of $\{1, \ldots, n\}$. The coefficient that corresponds to Z is: $\hat{t}(Z) =< t, \chi_Z >= 2^{-n}\sum_{x\in\{0,1\}^n} t(x)\chi_Z(x)$. The conjunction $t(x) \neq 0$ only when all its literals are satisfied, so we are interested in the values of χ only in that region. Let $ind(t)$ denote the indices of the variables that appear in t. We have two cases with respect to Z. If $Z \subseteq ind(t)$, then $\chi_Z(x)$ has a fixed value when $t(x) = 1$ (either -1 or 1), and $\hat{t}(Z)$ is $2^{-n+n-k} = 2^{-k}$ or -2^{-k} depending on that value. If $Z \not\subseteq ind(t)$, let $i \in Z \setminus ind(t)$, then $\forall x \in \{0,1\}^n$ we have that $t(x) = t(x^i)$ and $\chi_Z(x) = -\chi_Z(x^i)$, where x^i means x with the i'th bit flipped. This implies $\hat{t}(Z) = 0$. ■

Theorem 2 *The class of Disjoint DNF expressions is learnable with membership queries with respect to the uniform distribution.*

Proof: Let f be a disjoint DNF expression with m terms. We use the result that terms longer than $\log(2m/\epsilon)$ can be thrown away if we use the uniform distribution, incurring prediction error of at most $\epsilon/2$ [6]. Let $g = t_1 \vee t_2 \vee \ldots \vee t_q$ be the function we get when we do that. The function g is still boolean and therefore $Prob_D[f(x) \neq g(x)] = E_D[(f-g)^2]$, and it has a small squared error. On the other hand g has at most m terms each with at most $\log(2m/\epsilon)$ literals, and as it is disjoint, it can be written as $g = t_1 + t_2 + \ldots + t_q$ (replacing the OR operator with the PLUS operator). That means that a coefficient $\hat{g}(Z)$ of g is $\hat{g}(Z) = \sum_{j=1}^{q}\hat{t_j}(Z)$, and from Lemma 2 we get that the number of non-zero coefficients of g is bounded by $m \cdot (2m/\epsilon)$. Theorem 1 implies the polynomial learnability. ■

Theorem 3 *The class of Disjoint* $\log n$ *DNF expressions is exactly learnable with membership queries.*

Proof: Let f be a disjoint DNF expression, with m terms, where each term is not longer than $\log n$. The disjointness property combined with Lemma 2 imply that f is $mn-sparse$, and Lemma 1 implies the learnability. ■

We note again that the class of disjoint DNF functions strictly includes the class of decision trees, and the class Disjoint $\log n$ DNF includes the class of depth $\log n$ decision trees, so the above results generalize the results from [5]. We further note that the learnability result in [5] is shown using a bound on a measure, called L_1, of decision trees (which is defined as $L_1(f) = \sum_Z |\hat{f}(Z)|$), whereas the proof we give here escapes the use of L_1.

Theorem 4 *The class of* $\log n$ *term DNF expressions is learnable with membership queries with respect to the uniform distribution.*

Proof: Let f be an m term DNF function. We use the identity $f \vee g = f + g - fg$ (where $\vee$ denotes OR, $+$ denotes PLUS, and $-$ denotes MINUS) to get the following (inclusion-exclusion) identity:

$$\begin{aligned} f &= t_1 \vee t_2 \vee \ldots \vee t_m \\ &= (t_1 + t_2 - t_1 t_2) \vee t_3 \vee \ldots \vee t_m \\ &= (t_1 + t_2 + t_3 - t_1 t_2 - t_1 t_3 - t_2 t_3 + t_1 t_2 t_3) \vee t_4 \ldots \vee t_m \\ &= \sum_{i_1} t_{i_1} - \sum_{i_1, i_2} t_{i_1} t_{i_2} + \ldots + \sum_{i_1, \ldots, i_m} (-1)^{m+1} t_{i_1} \ldots t_{i_m} \end{aligned}$$

The last expression for f is a sum of 2^m conjunctions. If $m = \log n$ then the sum includes only n conjunctions. Now use the same arguments as in Theorem 2 to get the learnability under the uniform distribution. ■

Blum and Rudich [2] have shown exact learnability of $\log n$ term DNF, so the last result is not new with respect to showing learnability of classes[1]. Nevertheless, the result gives a different characterization for this class and might be useful in understanding the power and structure of DNF expressions.

Acknowledgments

Eyal Kushilevitz was very helpful while studying this material, and in reading earlier versions of this paper. I would also like to thank Les Valiant for insightful comments.

References

[1] H. Aizenstein and L. Pitt. Exact learning of read-k disjoint dnf and not-so-disjoint dnf. In *Proceedings of the ACM Workshop on Computational Learning Theory '92*, pages 71–76, Pittsburgh, Pennsylvania, 1992. Morgan Kaufmann.

[2] A. Blum and S. Rudich. Fast learning of k-term DNF formulas with queries. In *Proceedings of the Twenty-Forth Annual ACM Symposium on Theory of Computing*, Victoria, British Columbia, Canada, 1992.

[3] N. Bshouty. Exact learning via the monotone theory. In *Proc. 23rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 302–311, 1993.

[4] R. Khardon, E. Kushilevitz, and D. Roth. Learning read $(\log n)^{1/3}$ disjoint DNF. Unpublished, July 1993.

[1] The learning algorithm in [2] uses equivalence queries and membership queries. The equivalence queries can be simulated using a random example oracle, and for the uniform distribution one can simply use random bits in combination with membership queries to simulate it.

[5] E. Kushilevitz and Y. Mansour. Learning decision trees using the fourier sprectrum. In *Proc. 23rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 455–464, 1991.

[6] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the ACM Workshop on Computational Learning Theory '90*, pages 314–326. Morgan Kaufmann, 1990.