Weakly Learning DNF and Characterizing Statistical Query Learning Using Fourier Analysis

Avrim Blum* Carnegie Mellon University

Merrick Furst* Carnegie Mellon University

Jeffrey Jackson* Carnegie Mellon University

Michael Kearns † AT&T Bell Laboratories

Yishay Mansour‡ Tel-Aviv University

Steven Rudich* Carnegie Mellon University

Abstract

We present new results, both positive and negative, on the well-studied problem of learning disjunctive normal form (DNF) expressions.

We first prove that an algorithm due to Kushilevitz and Mansour [16] can be used to weakly learn DNF using membership queries in polynomial time, with respect to the uniform distribution on the inputs. This is the first positive result for learning unrestricted DNF expressions in polynomial time in any nontrivial formal model of learning. It provides a sharp contrast with the results of Kharitonov [15], who proved that AC^0 is not efficiently learnable in the same model (given certain plausible cryptographic assumptions). We also present efficient learning algorithms in various models for the read-k and SAT-k subclasses of DNF.

For our negative results, we turn our attention to the recently introduced *statistical query* model of learning [11]. This model is a restricted version of the popular Probably Approximately Correct (PAC) model [23], and practically every class known to be efficiently learnable in the PAC model is in fact learnable in the statistical query model [11]. Here we give a general characterization of the complexity of statistical query learning in terms of the number of uncorrelated functions in the concept class. This is a distribution-dependent quantity yielding upper and lower bounds on the number of statistical queries required for learning on any input distribution. As a corollary, we obtain that DNF expressions and decision trees are not even weakly learnable with respect to the uniform input distribution in polynomial time in the statistical query model. This result is information-theoretic and therefore does not rely on any unproven assumptions. It demonstrates that no simple modification of the existing algorithms in the computational learning theory literature for learning various restricted forms of DNF and decision trees from passive random examples (and also several algorithms proposed in the experimental machine learning communities, such as the ID3 algorithm for decision trees [22] and its variants) will solve the general problem.

The unifying tool for all of our results is the Fourier analysis of a finite class of boolean functions on the hypercube.

1 Introduction and History

We present new results, both positive and negative, on the well-studied problem of learning DNF expressions. The problem of efficiently learning DNF in any nontrivial formal model of learning has been of central interest to the computational learning theory community since the seminal paper of Valiant [23] introducing the popular Probably Approximately Correct (PAC) learning model. Despite the importance of this problem, prior to this work no polynomial-time algorithm for learning unrestricted DNF had been discovered, even when the uniform input distribution was considered, and even when the learning algorithm was allowed to make membership queries and output a hypothesis only slightly outperforming random guessing (known as *weak* learning). Indeed, in the distribution-independent PAC model there are representation-dependent hardness results (that is, hardness results that assume certain syntactic restrictions on the learning algorithm's hypothesis) for learning even some rather restricted forms of DNF [21, 12]. These hardness results left unresolved the status of learning DNF formulas in the absence of hypothesis restrictions, or with respect to the uniform distribution, or using membership queries.

We prove that an algorithm due to Kushilevitz and Mansour [16] can be used to weakly learn DNF formulas with respect to the uniform distribution using membership queries. Thus, we give the first positive result for learning unrestricted DNF in a nontrivial model of learning. This result provides a sharp contrast between DNF formulas and the more general class of AC^0 circuits, which Kharitonov [15] proved is not learnable in this same model under certain cryptographic assumptions.

Due to the dearth of positive results for unrestricted DNF expressions, various restricted DNF classes have attracted considerable attention in the literature [4, 2, 10, 3, 1, 5, 17, 8]. Here we extend some of these results. In particular,

---

*This research is sponsored in part by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and the Advanced Research Projects Agency (ARPA) under grant number F33615-93-1-1330. Support also is sponsored by the National Science Foundation under Grant No. CC-9119319. Blum also supported in part by NSF National Young Investigator grant CCR-9357793. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of Wright Laboratory or the United States Government, or NSF.

†Contact author. Address: AT&T Bell Laboratories, Room 2A-423, 600 Mountain Avenue, P.O. Box 636, Murray Hill, NJ 07974. Electronic mail: mkearns@research.att.com

‡This research was supported in part by The Israel Science Foundation administered by The Israel Academy of Science and Humanities and by a grant of the Israeli Ministry of Science and Technology.

it is known that the class of read-k DNF (that is, DNF expressions in which every variable appears at most k times) is learnable in polynomial time in the distribution-independent PAC model using membership queries for $k \leq 2$ [2, 10], but is as hard to learn in this same model as unrestricted DNF for $k \geq 3$ [10]. Aizenstein and Pitt [3] have shown that read-k, SAT-ℓ DNF (that is, DNF expressions which are both read-k and such that at most ℓ terms are satisfied by any input) can be efficiently learned in the distribution-independent PAC model using membership queries.

Here we show that, using membership queries and with respect to the uniform input distribution, read-k DNF is learnable in polynomial time with an accuracy that is a constant depending on k. We also prove that SAT-k DNF is strongly learnable in time exponential in k (but otherwise polynomial), and that SAT-k, $\log(n)$-DNF is exactly learnable using membership queries in the same time bound.

Finally, we examine the learnability of a class that contains many functions, including parity and majority, which are not in DNF. Let $\widehat{PT}_1$ denote the class of functions that are computable as a majority of a polynomial number of parities. We show that $\widehat{PT}_1$ is weakly learnable with respect to uniform using membership queries.

In the second part of the paper, we examine the DNF learning problem in the recently introduced *statistical query* model of learning [11]. This is a restricted version of the PAC model in which the learning algorithm does not actually receive labeled examples of the unknown target function drawn with respect to the input distribution. Instead, in the statistical query model the learner may specify any *property* of labeled examples, and obtain accurate estimates of the probability that a random example will possess the property. An important feature of this model is that any class efficiently learnable from statistical queries (for all distributions or special distributions, respectively) is efficiently learnable in the PAC model with an arbitrarily high rate of classification noise (for all distributions or special distributions, respectively) [11]. Furthermore, it has been demonstrated [11] that practically every class known to be efficiently learnable in the PAC model (either for all distributions or special distributions) is also efficiently learnable in the statistical query model (and thus is efficiently PAC learnable with classification noise). In other words, PAC model algorithms almost always learn by estimating probabilities. (A notable exception to this is the class of parity functions, which is known to be efficiently learnable in the PAC model but is not efficiently learnable in the statistical query model [11].) Consequently, any light we can shed on the problem of learning DNF expressions in the statistical query model provides insight into the still unresolved problem of learning DNF in the basic PAC model.

Here we provide a general characterization of the number of statistical queries required for learning that is applicable to any concept class with respect to any input distribution. We prove that if a class contains a superpolynomial number of nearly uncorrelated functions with respect to the input distribution, then a superpolynomial number of statistical queries are required for learning with respect to that distribution. On the other hand, if a class contains only a polynomial number of nearly uncorrelated functions with respect to the distribution, then the class is weakly learnable by a (possibly non-uniform) polynomial time algorithm.

A corollary of this characterization is that DNF formulas and decision trees are not even weakly learnable in polynomial time with respect to the uniform distribution in the statistical query model. This result does not rely on any unproven assumptions. An important consequence of this result is that no algorithm which can be cast as a statistical query algorithm can learn unrestricted forms of DNF and decision trees. Therefore, no simple modification of the existing algorithms from the computational learning theory literature for learning various restricted forms of DNF and decision trees from passive random examples will solve the general problem. The same statement also applies to several well-studied algorithms proposed in the experimental machine learning community, including the ID3 algorithm for learning decision trees [22] and its variants.

All of our results rely heavily on the Fourier representation of functions on the hypercube [18, 16, 20], demonstrating once again the utility of spectral analysis tools in computational learning theory.

2 Definitions and Notation

2.1 Learning on the Uniform Distribution Using Membership Queries

A *concept* is a boolean function on an *input space* X (which in this paper will always be $\{0,1\}^n$), and for convenience we define boolean functions to have outputs in $\{+1,-1\}$. A *concept class* $\mathcal{F}$ is a set of concepts. An *input* $\vec{x}$ is an element of the input space $\{0,1\}^n$. We use x_i to denote the ith bit of $\vec{x}$ and we generally use f to denote the chosen *target concept* from $\mathcal{F}$.

We say that a (possibly randomized) function g is an *ϵ-approximation* of f if [1] $\mathbf{Pr}[f = g] \geq 1 - \epsilon$, where the probability is taken over the uniform distribution on the input space and over any random choices made by g.

A *membership query* is a query to an oracle for f for the value of f on a desired input $\vec{x}$. If there is an algorithm $\mathcal{A}$ with access to membership queries and such that for any positive ϵ and δ and any target concept $f \in \mathcal{F}$, with probability at least $1-\delta$ algorithm $\mathcal{A}$ produces as output an ϵ-approximation for f in time polynomial in n, the size s of f, $1/\epsilon$, and $1/\delta$, then we say that $\mathcal{F}$ is *(strongly) learnable using membership queries with respect to the uniform distribution.* The *size* of a concept f is a measure of the number of bits in the smallest representation of f; throughout this paper we will use the number of terms in the smallest DNF representation of f as the size s of f. The parameters ϵ and δ above are called the *accuracy* and *confidence* of the approximation, respectively.

If there is a polynomial $p(n,s)$ and an algorithm $\mathcal{A}$ with access to membership queries such that for any positive δ and any target concept $f \in \mathcal{F}$, with probability at least $1-\delta$ algorithm $\mathcal{A}$ produces as output a $1/2 - 1/p(n,s)$-approximation for f in time polynomial in n, s, and $1/\delta$, then $\mathcal{F}$ is *weakly learnable using membership queries with respect to the uniform distribution*, or alternatively, is *efficiently $1/2 - 1/p(n,s)$-approximated.*

2.2 The Statistical Query Learning Model

Unlike the membership query models of learning we have defined so far, in the *statistical query* learning model [11] the learner is *not* explicitly allowed to see labeled examples

[1] Unless subscripted by a distribution D, all probabilities and expectations are taken with respect to the uniform distribution on $\{0,1\}^n$.

$(\vec{x}, f(\vec{x}))$ of the target concept, but instead may only *estimate probabilities* involving labeled examples. We formalize this as follows: the learning algorithm is given access to a *statistics oracle*. A query to this oracle is a pair (g, τ), where g is a function $g : \{0,1\}^n \times \{+1,-1\} \to \{+1,-1\}$, and $\tau \in [0,1]$ is a real number called the *tolerance* of the query. The oracle may respond to the query (g, τ) with any value v satisfying

$$\mathbf{E}_D[g(\vec{x}, f(\vec{x}))] - \tau \le v \le \mathbf{E}_D[g(\vec{x}, f(\vec{x}))] + \tau.$$

Since we will examine statistical query learnability not just with respect to the uniform input distribution, but with respect to any fixed distribution D on $\{0,1\}^n$, we have placed a subscript on the expectation indicating the distribution.

Thus, a statistical query algorithm may obtain estimates for the expectations of binary-valued random variables of its own choosing. Note that if we regard the query function g as computing a property of labeled examples (with $g(\vec{x}, f(\vec{x})) = +1$ indicating that the property holds), then the response to the query provides the learning algorithm with an estimate for the probability that the property holds that is accurate within additive error τ.

We say that the concept class $\mathcal{F}$ is *learnable from statistical queries* with respect to an input distribution D if there is a learning algorithm $\mathcal{A}$ with access to statistical queries for the target function and input distribution, such that for any positive ϵ and any target $f \in \mathcal{F}$, algorithm $\mathcal{A}$ produces an ϵ-approximation for f (with respect to D) in time polynomial in n, the size of f, and $1/\epsilon$. Furthermore, algorithm $\mathcal{A}$ must only make queries (g, τ) in which g can be computed by a circuit whose size is bounded by a fixed polynomial in the parameters, and in which τ is lower bounded by the inverse of a fixed polynomial in the parameters[2]. Thus, $\mathcal{A}$ must run in polynomial time, and is allowed to make only efficiently computable queries with inverse polynomial tolerance.

The motivation for this notion of efficiency is that every class learnable from statistical queries is efficiently learnable in the PAC model by a straightforward simulation argument [11], and thus the statistical query model can be regarded as a natural *restriction* on the type of computations a PAC model algorithm can perform. The general motivation for the statistical query model can be found in the paper of Kearns [11]. Here it suffices to reiterate that almost every class known to be efficiently learnable in the PAC model or its distribution-specific variant can be shown to be learnable in the statistical query model, and furthermore statistical query learning implies efficient PAC learning even in the presence of large amounts of classification noise.

2.3 DNF Expressions

A DNF formula is a disjunction of terms, where each term is a conjunction of literals and a literal is either a variable or its negation. For a given DNF formula f we use s to denote the number of terms in f, T_i to represent the ith term in f (the ordering is arbitrary), and V_i to denote the set of variables in T_i. A DNF formula f is *k-DNF* if it has at most k literals in each term, is *read-k* if each variable appears at most k times, and is *SAT-k* if no input satisfies more than k terms of f.

[2] Note that allowing queries with $\tau = 0$ would provide the algorithm with at least the power of membership queries.

We assume for convenience that the `true` output value of a boolean function is represented by $+1$ and the `false` value by -1.

2.4 The Fourier Transform

For each bit vector $\vec{a} \in \{0,1\}^n$ we define the function $\chi_{\vec{a}} : \{0,1\}^n \to \{+1,-1\}$ as

$$\chi_{\vec{a}}(\vec{x}) = (-1)^{\sum_{i=1}^n a_i x_i} = 1 - 2\left(\sum_{i=1}^n a_i x_i \bmod 2\right).$$

That is, let $\vec{u}_{\vec{a},\vec{x}}$ represent the vector of bits from $\vec{x}$ for which the corresponding bits in $\vec{a}$ are 1. Then $\chi_{\vec{a}}(\vec{x})$ is the boolean function that is 1 when the parity of $\vec{u}_{\vec{a},\vec{x}}$ is even and is -1 otherwise. Defined this way, the 2^n parity functions $\chi_{\vec{a}}$ have a number of useful properties which we will exploit repeatedly.

First, with inner product defined by $\langle f, g\rangle = \mathbf{E}[fg]$ and norm by $\|f\| = \sqrt{\mathbf{E}[f^2]}$, $\{\chi_{\vec{z}}\}_{\vec{z}\in\{0,1\}^n}$ is an orthonormal basis for the vector space of real-valued functions on the Boolean cube $\mathbf{Z}_2^n$. That is, every function $f : \{0,1\}^n \to \mathbf{R}$ can be uniquely expressed as a linear combination of parity functions:

$$f = \sum_{\vec{a}\in\{0,1\}^n} \hat{f}(\vec{a})\chi_{\vec{a}}.$$

We call the vector of coefficients $\hat{f}$ the *Fourier transform* of f. Because of the orthonormality of the parity functions, $\hat{f}(\vec{a}) = \mathbf{E}[f\chi_{\vec{a}}]$. Thus for boolean f, $\hat{f}(\vec{a})$ represents the correlation of f and $\chi_{\vec{a}}$. Also note that $\hat{f}(\vec{0}) = \mathbf{E}[f\chi_{\vec{0}}] = \mathbf{E}[f]$. We call $\hat{f}(\vec{0})$ the *constant Fourier coefficient* since $\chi_{\vec{0}}$ is the constant function $+1$. Finally, the Fourier transform is a linear operator. That is, if $h = cf + g$ for functions f, g and scalar c, then $\hat{h} = c\hat{f} + \hat{g}$.

Parseval's identity states that for every function f, $\mathbf{E}[f^2] = \sum_{\vec{a}} \hat{f}^2(\vec{a})$. For boolean f it follows that $\sum_{\vec{a}} \hat{f}^2(\vec{a}) = 1$, a fact we use frequently.

At times we use a subset A of the n variables of a function as the index of a parity or Fourier coefficient, with the following meaning: χ_A denotes the function $\chi_{\vec{a}}$ where $\vec{a}$ is the characteristic vector corresponding to A, and $\hat{f}(A)$ has a similar interpretation.

A *t-sparse function* is a function that has at most t nonzero Fourier coefficients. The *support* of a function f is the set $\{A \mid \hat{f}(A) \neq 0\}$.

3 Preliminaries

Our positive learnability results rely heavily on an algorithm of Kushilevitz and Mansour [16] (the *KM algorithm*) which finds, with high probability, close approximations to all of the large Fourier coefficients of a function f. The KM algorithm is allowed to make membership queries for f. Kushilevitz and Mansour have shown that given such approximate coefficients one can strongly learn some important concept classes such as decision trees [16]. However, while the KM algorithm is a key element of our learning scheme, we need to extend their approach somewhat to handle the case where the large Fourier coefficients give us only a weak approximation to the target function.

The main idea behind our positive results is to show that DNF formulas have enough sufficiently large Fourier coefficients that the KM algorithm can be usefully applied. We then use a general transformation that shows how to take a deterministic approximation g which is significantly (that is, inverse polynomially) closer to f than the origin (regarding the functions as vectors), and produce a randomized approximation h such that $\mathbf{Pr}[f \neq h]$ is similarly better than $1/2$. Thus our learning problem reduces to finding a function g appropriately "close to" f. To show that the KM algorithm can find such a g for the concept classes we consider, we will combine known results about the KM algorithm with a new bound on the size of Fourier coefficients for functions in these classes.

We begin by stating as a lemma the known results about the KM algorithm which we will need. These and the other results of this section hold for any class of boolean functions, not just DNF.

Lemma 1 (Kushilevitz & Mansour) *For any target concept f, threshold θ, and $\epsilon, \delta > 0$, the KM algorithm, with probability at least $1-\delta$, outputs all the nonzero Fourier coefficients of a function g whose support S obeys the following properties:*

1. *S contains every set A such that $|\hat{f}(A)| > \theta$.*
2. $\displaystyle\sum_{A \in S} (\hat{f}(A) - \hat{g}(A))^2 \leq \epsilon.$
3. *$|S|$ is polynomial in $1/\theta$.*

The algorithm uses membership queries, and runs in time polynomial in n, $1/\theta$, $1/\epsilon$, and $\log(1/\delta)$.

We use $KM(\theta, \epsilon, \delta)$ to represent an execution of the KM algorithm with the respective threshold, accuracy, and confidence parameters. That it is possible for the algorithm to return a number of coefficients that is polynomial in $1/\theta$ follows from the fact that since $\sum_A \hat{f}^2(A) = 1$, there are at most $1/\theta^2$ coefficients of f with magnitude at least θ.

We now turn to bounding the difference between the target f and the function g returned by the KM algorithm. It can be shown that for boolean target functions f, running $KM(\theta, \epsilon, \delta)$ produces a function g (with support S) that with probability at least $1-\delta$ has $\mathbf{E}[(f-g)^2] = \epsilon + 1 - \sum_{A \in S} \hat{f}^2(A)$. To see this, note that we may write:

$$
\begin{aligned}
\mathbf{E}[(f-g)^2] &= \sum_A \widehat{(f-g)}^2(A) \\
&= \sum (\hat{f}(A) - \hat{g}(A))^2 \\
&\leq \epsilon + \sum_{A \notin S} (\hat{f}(A) - \hat{g}(A))^2 \\
&= \epsilon + 1 - \sum_{A \in S} \hat{f}^2(A).
\end{aligned}
$$

The first equality is Parseval, the second is by linearity of the Fourier transform, and the inequality is by Lemma 1. The final equality holds because $\hat{g}(A) = 0$ for $A \notin S$, and $\sum_A \hat{f}^2(A) = 1$.

While the Kushilevitz and Mansour analysis assumes that the summation in this final bound is near 1, for our weak learning results the value we assume for this summation may be quite small (but non-negligible). The following lemma gives us a bound on $\mathbf{E}[(f-g)^2]$ in terms of a lower bound on $\sum_{A \in S} \hat{f}^2(A)$.

Lemma 2 *Given a $\{+1,-1\}$-valued function f, let S be a set such that $\sum_{A \in S} \hat{f}^2(A) \geq \alpha$, and let g be the output of $KM(\sqrt{\alpha/(4|S|)}, \alpha/4, \delta)$. Then with probability at least $1-\delta$, $\mathbf{E}[(f-g)^2] \leq 1 - \alpha/2$.*

Proof: Let T be the support of g. Then with probability at least $1-\delta$, for all $A \in S - T$, $\hat{f}(A) \leq \sqrt{\alpha/(4|S|)}$ and thus $\sum_{A \in S-T} \hat{f}^2(A) \leq \alpha/4$. Therefore with at least this probability

$$
\begin{aligned}
\mathbf{E}[(f-g)^2] &\leq 1 + \alpha/4 - \sum_{A \in T} \hat{f}^2(A) \\
&\leq 1 + \alpha/4 - \left(\sum_{A \in S} \hat{f}^2(A) - \sum_{A \in S-T} \hat{f}^2(A) \right) \\
&\leq 1 - \alpha/2.
\end{aligned}
$$

□

Now we are ready to link the squared error measure above with the notion of ϵ-approximation.

Lemma 3 *Given a $\{+1,-1\}$-valued function f and a deterministic approximation g, define the randomized function h as follows: let $h(\vec{x}) = -1$ with probability*

$$p = \frac{(1 - g(\vec{x}))^2}{2(1 + g^2(\vec{x}))}$$

and $h(\vec{x}) = 1$ with probability $1-p$. Then $\mathbf{Pr}[f \neq h] \leq (1/2)\mathbf{E}[(f-g)^2]$. So, if $\mathbf{E}[(f-g)^2] \leq 1 - \alpha$ then h is a $1/2 - \alpha/2$-approximation for f.

Proof: First, the algorithm is well-defined since $0 \leq p \leq 1$ for any value of $g(\vec{x})$. Noting that $1 - p = \mathbf{Pr}[h(\vec{x}) = 1]$ can be written as $(-1 - g(\vec{x}))^2/2(1 + g^2(\vec{x}))$, it follows that for any fixed $\vec{x}$, $\mathbf{Pr}[h(\vec{x}) \neq f(\vec{x})] = (f(\vec{x}) - g(\vec{x}))^2/2(1 + g^2(\vec{x}))$, where the probability is taken over the random choices made by h. Now considering the distribution over all inputs $\vec{x}$ as well as h's random choices, we get

$$\mathbf{Pr}[h \neq f] \leq \frac{1}{2}\mathbf{E}[(f-g)^2].$$

□(Lemma 3)

A similar but slightly weaker randomized approximation method was given by Kearns, Schapire, and Sellie [13]. Putting the results of this section together, we have the following.

Theorem 4 *A concept class $\mathcal{F}$ is weakly learnable with membership queries with respect to the uniform distribution if there are polynomials p and q such that for every $f \in \mathcal{F}$ there is a set S with $|S| \leq p(n,s)$ such that $\sum_{A \in S} \hat{f}^2(A) \geq 1/q(n,s)$, where s represents the size of f. In particular, for every f in such a class the algorithm*

$$KM\left(\frac{1}{2\sqrt{p(n,s)q(n,s)}}, \frac{1}{4q(n,s)}, \delta \right)$$

plus the approximation scheme of Lemma 3 will with probability at least $1-\delta$ *produce a randomized* $1/2 - 1/4q(n,s)$*-approximation of* f*. The algorithm runs in time polynomial in* n*,* s*, and* $\log(1/\delta)$*.*

4 Positive Results

4.1 Weakly Learning DNF

Linial, Mansour, and Nisan [18] showed that AC^0, the class of constant-depth circuits, is learnable in superpolynomial but subexponential time with respect to the uniform distribution by proving that for every AC^0 function f almost all of the "large" Fourier coefficients of f are coefficients of parities of "few" variables. We show that an even stronger property holds for the Fourier transform of any DNF function, a property which will be key to several of our positive results about DNF learnability. The following definition will simplify the statement and proof of this property.

Definition 1 *Let* f *be a DNF formula and let* T_i *(with variables* V_i*) be a term in* f*. Then for every* $A \subseteq V_i$*, define* $\chi_A(T_i)$ *to be* $\chi_A(\vec{x})$*, where* $\vec{x}$ *is any input which satisfies* T_i*.*

Lemma 5 *Let* f *be a DNF formula. Then for every term* T_i *(with variables* V_i*),*

$$\sum_{A \subseteq V_i} \hat{f}(A)\chi_A(T_i) = +1.$$

Proof: Consider a particular term T_i of f. Let f_i represent the restriction of f obtained by fixing the variables in V_i so that T_i is satisfied. Then $f_i \equiv +1$. Since $\chi_{\vec{0}} \equiv +1$, $\hat{f}_i(\vec{0}) = \mathbf{E}[f_i\chi_{\vec{0}}] = 1$. Now since $f = \sum \hat{f}(A)\chi_A$, the restriction f_i is also a linear combination of the restrictions $\chi_{A,i}$ of the χ_A's obtained by fixing the variables in V_i as above, that is,

$$f_i = \sum_{A \subseteq \{x_1,\ldots,x_n\}} \hat{f}(A)\chi_{A,i}.$$

For all $A \subseteq V_i$, $\chi_{A,i} = \chi_A(T_i)$ is a constant function. On the other hand, for all $A \not\subseteq V_i$, the restriction $\chi_{A,i}$ is not a constant since some variables in χ_A survive the restriction. Thus $\hat{f}_i(\vec{0}) = \sum_{A \subseteq V_i} \hat{f}(A)\chi_A(T_i)$, and as established above, $\hat{f}_i(\vec{0}) = 1$. □(Lemma 5)

A particularly useful implication of the lemma for our purposes is that for every term T_i in f, there is some $A \subseteq V_i$ such that $|\hat{f}(A)| \geq 2^{-|V_i|}$. Thus if even one term in a DNF f has $O(\log s)$ variables then there is at least one Fourier coefficient of f which is inverse polynomially large. This allows us to use the KM algorithm to weakly learn DNF with membership queries with respect to the uniform distribution.

Theorem 6 *The class of DNF formulas can be efficiently* $(1/2 - 1/6s)$*-approximated.*

Proof: We assume that there is at least one term in f with at most $\log(3s)$ literals; otherwise, f is sufficiently well-approximated by the constant -1 function. Thus by Lemma 5 there is at least one Fourier coefficient (call if $\hat{f}(A)$) of magnitude $1/3s$. The parity χ_A corresponding to $\hat{f}(A)$ can be found with probability $1-\delta$ in time polynomial in n, s, and $\log(1/\delta)$ by $KM(1/3s, 1, \delta)$. As $\hat{f}(A)$ represents the correlation of χ_A and f, $g = \text{sign}(\hat{f}(A))\chi_A$ is an adequate approximation. □(Theorem 6)

A related but more complicated algorithm yields improved accuracy (the details of the algorithm are omitted).

Theorem 7 *The class of DNF formulas can be efficiently* $(1/2 - \Omega(\log(s)/s))$*-approximated.*

4.2 Learning Read-k DNF

Lemma 5 gives us that every term has at least one "large" Fourier coefficient associated with it. However, conceivably a small set of large coefficients are shared by many of the terms, so there may be very few large coefficients in the DNF formula. On the other hand, each coefficient (except the constant coefficient) of a read-k formula may be shared by at most k terms. We use this fact to obtain an accuracy bound of $1/2 - \Omega(1/k)$ for the class of read-k DNF.

Theorem 8 *For every* k*, the class of read-*k *DNF can be efficiently* $(1/2 - 1/16k)$*-approximated.*

Proof: For any read-k DNF f we will show that there is a set S with $|S| \leq 24n^2k^2$ such that $\sum_{A\in S} \hat{f}^2(A) \geq 1/4k$. The result then follows from Theorem 4.

To derive this bound, first consider the case $k = 1$. Lemma 5 implies that for each term T_i: $\sum_{A\in 2^{V_i}} |\hat{f}(A)| \geq 1$ where 2^{V_i} represents the power set of V_i. Define $S = \cup_i 2^{V_i}$. Because $k = 1$, for any $i \neq j$, $2^{V_i} \cap 2^{V_j} = \{\emptyset\}$. Thus, letting S_i denote the set $2^{V_i} - \emptyset$,

$$\sum_{A\in S} \hat{f}^2(A) \geq \sum_i \sum_{A \in S_i} \hat{f}^2(A).$$

We will assume that $|\hat{f}(\vec{0})| \leq 1/6$, since otherwise f is adequately approximated by a constant function. Thus for each T_i, $\sum_{A\in S_i} |\hat{f}(A)| \geq 5/6$, which implies that $\sum_{A\in S_i} \hat{f}^2(A) \geq (5/6)^2/|S_i|$. So,

$$\sum_{A\in S} \hat{f}^2(A) \geq \left(\tfrac{5}{6}\right)^2 \sum_i \frac{1}{2^{|V_i|}-1}.$$

By the restriction on $\hat{f}(\vec{0})$ we know that at least $5/12$ of the inputs satisfy f. Since the fraction of inputs which satisfy a term T_i is $2^{-|V_i|}$, $\sum_i 2^{-|V_i|} \geq 5/12$ and so $\sum_{A\in S} \hat{f}^2(A) \geq (5/6)^2(5/12) > 1/4$.

Now consider larger k. In this case, for any given set A we can have $A \in S_i$ for up to (but no more than) k distinct values of i. Thus

$$\sum_{A\in S} \hat{f}^2(A) \geq \frac{1}{k} \sum_i \sum_{A\in S_i} \hat{f}^2(A)$$

and therefore $\sum_{A\in S} \hat{f}^2(A) > 1/4k$.

Finally, we need to bound $|S|$. In general, the set S above can be exponentially large even for a read-once DNF. We get around this by considering only "small" terms when constructing S. Specifically, we now let

$$S = \bigcup_{|V_i| \leq \log(24kn)} 2^{V_i}.$$

Because there are at most kn terms in a read-k DNF, the terms which are excluded from S are satisfied by at most $1/24$ of the inputs. The included terms are therefore satisfied by at least $5/12 - 1/24 = 9/24$ of the inputs, and using this value rather than $5/12$ in the earlier analysis still gives the desired bound. □(Theorem 8)

4.3 Learning SAT-k DNF

We demonstrate the (strong) learnability of SAT-k DNF for constant k by showing that every SAT-k DNF is well-approximated by a function with small support.

Theorem 9 [3] *For any k, the class of SAT-k DNF formulas can be ϵ-approximated by a randomized learning algorithm which uses membership queries, succeeds with probability $1-\delta$, and runs in time polynomial in n, s^k, $1/\epsilon^k$, and $\log(1/\delta)$.*

Proof: We will show that there is some polynomially sparse deterministic function g such that $\mathbf{E}[(f-g)^2] \leq \epsilon/2$. The result then follows from standard arguments.

Let $r = 8s/\epsilon$ and let g be what remains of f after removing any terms having more than $\log(r)$ variables. Then $\mathbf{E}[(f-g)^2] = 4\mathbf{Pr}[f \neq g] \leq \epsilon/2$. The inequality holds because each term removed from f covers at most an $\epsilon/8s$ fraction of the input space. To see that g has small support, let s' represent the number of terms in g and define $P_i(\vec{x})$, $1 \leq i \leq s'$, to be $+1$ if $\vec{x}$ satisfies the ith term of g and 0 if $\vec{x}$ does not. At most $\log(r)$ variables are relevant for P_i and thus the Fourier representation of P_i has no more than r non-zero coefficients. Using the principle of inclusion-exclusion we can create a function P' from the P_i's which is $(rs')^k$-sparse and which is 1 when g is satisfied and 0 otherwise. Specifically, let

$$P' = \sum_{i_1} P_{i_1} - \sum_{i_1<i_2} P_{i_1}P_{i_2} + \sum_{i_1<i_2<i_3} P_{i_1}P_{i_2}P_{i_3} - \cdots - (-1)^k \sum_{i_1<\cdots<i_k} P_{i_1}\cdots P_{i_k}.$$

It can be verified inductively that this polynomial has the claimed properties. Noting that $g = 2P' - 1$ completes the proof. □(Theorem 9)

By restricting the size of terms in the SAT-k DNF's considered and using exact reconstruction and derandomization techniques similar to those of Kushilevitz and Mansour [16], we can extend the above to a deterministic, distribution-independent learning result (this generalizes a similar result for SAT-1 (disjoint) DNF by Khardon [14]).

Theorem 10 *For any k, the class of SAT-k $O(\log s)$-DNF formulas of s terms can be learned exactly by a deterministic learning algorithm which uses membership queries and runs in time polynomial in n and s^k.*

4.4 Learning $\widehat{PT}_1$

In this section we generalize our weak learning result for unrestricted DNF. In particular, let $\widehat{PT}_1$ represent the class of functions computable as the majority of a polynomial (in n) number of parities; equivalently, this is the class of functions that are the sign of a polynomially-sparse function having polynomially-bounded integer Fourier coefficients. We show that $\widehat{PT}_1$ is weakly learnable with respect to uniform using queries. $\widehat{PT}_1$ is a rather general class containing many functions, such as majority, that are not approximable by AC^0 circuits. Our weak learnability proof builds on the work of Bruck [7].

Theorem 11 *$\widehat{PT}_1$ is weakly learnable using membership queries with respect to the uniform distribution.*

Proof: By definition, for any $f \in \widehat{PT}_1$ there is some $g = \sum_{A\in S} \hat{g}(A)\chi_A$ such that $f = \mathrm{sign}(g)$, $|S| \leq p(n)$, and for all $A \in S$ we have both that $\hat{g}(A)$ is an integer and $|\hat{g}(A)| \leq p(n)$ for some polynomial p. Since $f = \mathrm{sign}(g)$,

$$\mathbf{E}[|g|] = \mathbf{E}[fg] = \sum_{A\in S} \hat{f}(A)\hat{g}(A),$$

where the final equality follows from a generalization of Parseval's identity and the fact that $\hat{g}(A) = 0$ for $A \notin S$. We can assume without loss of generality that for all $x \in \{0,1\}^n$, $g(x) \neq 0$, and therefore $\mathbf{E}[|g|] \geq 1$. Thus for some $A \in S$, $|\hat{f}(A)| \geq 1/p^2(n)$, and we can therefore use KM to find a weak approximator for f. □(Theorem 11)

5 Characterizing Statistical Query Learning

In this second part of the paper, we present results that characterize when a given class of functions is weakly learnable under a given distribution in the statistical query model. An important corollary of this characterization is that the class of parity functions on $\log n$ variables (that is, the class of functions χ_A where $|A| = O(\log n)$) cannot be even weakly learned with a polynomial number of statistical queries (each with inverse polynomial tolerance), even with respect to the uniform input distribution. This immediately implies that DNF expressions and decision trees, both of which contain the $\log n$-bit parities as a subclass, are not weakly learnable in the statistical query model, even with respect to the uniform input distribution.

Our lower bound is information-theoretic, and thus does not rely on any unproven assumptions. For our upper bound we actually give a (non-uniform) *polynomial time* weak learning algorithm. Thus, we find that in the statistical query model, weak learnability from a polynomial number of queries of inverse polynomial tolerance, but in *any* amount of computation time, in fact implies *efficient* (although possibly non-uniform) weak learnability. This is quite different from the situation in the PAC model, where information-theoretic learnability provably does not imply (even non-uniform) polynomial time learnability given certain (non-uniform) cryptographic assumptions.

In order to present our characterization, we need the following key definition.

Definition 2 *For $\mathcal{F}$ a class of boolean functions over $\{0,1\}^n$ and D a distribution over $\{0,1\}^n$, we define SQ-DIM$(\mathcal{F}, D)$, the* statistical query dimension *of $\mathcal{F}$ with respect to D, to be the largest natural number d such that $\mathcal{F}$ contains d functions $f_1, \ldots, f_d$ with the property that for all $i \neq j$ we have:*

$$|\mathbf{Pr}_D[f_i = f_j] - \mathbf{Pr}_D[f_i \neq f_j]| \leq \frac{1}{d^3}.$$

[3] A similar result has also been shown by Lipton using a somewhat different analysis [19].

Thus, if SQ-DIM$(\mathcal{F}, D) = d$ it means that there are d "nearly uncorrelated" functions in the class $\mathcal{F}$ (with respect to the distribution D). Note that unlike the well-known Vapnik-Chervonenkis (VC) dimension, which is a distribution-independent quantity and is known to characterize the number of random examples required to learn in the distribution-independent PAC model [6], the statistical query dimension is a distribution-dependent quantity. It is possible to prove a one-sided polynomial relationship between the two quantities: namely, if $\mathcal{F}$ is a class of VC dimension d, then there exists a distribution D such that SQ-DIM$(\mathcal{F}, D) = \Omega(d)$ [4]. However, there is no such polynomial relationship in the other direction, as there are function classes and distributions whose statistical query dimension may be exponential in the VC dimension of the function class (for instance, the class of all parity functions on $\{0,1\}^n$ with respect to the uniform input distribution).

We now state the main theorems of this section, which establish that the statistical query dimension characterizes (within a polynomial factor) the number of statistical queries that must be made for learning.

Theorem 12 *(Lower Bound) Let $\mathcal{F}$ be a class of functions over $\{0,1\}^n$ and D a distribution such that SQ-DIM$(\mathcal{F}, D) \geq d \geq 16$. Then if all queries are made with a tolerance of at least $1/d^{1/3}$, at least $d^{1/3}/2$ queries are required to learn $\mathcal{F}$ with error less than $1/2 - 1/d^3$ in the statistical query model.*

Theorem 13 *(Upper Bound) If $\mathcal{F}$ is a class of functions over $\{0,1\}^n$ and D a distribution such that SQ-DIM$(\mathcal{F}, D) = d$, then there is a learning algorithm for $\mathcal{F}$ with respect to D in the statistical query model that makes d queries, each of tolerance at least $1/3d^3$, and finds a hypothesis with error at most $1/2 - 1/3d^3$.*

We shall give the proofs for these main theorems momentarily.

If we think of $\mathcal{F}$ and D as function and distribution *ensembles* (one for each n), then Theorems 12 and 13 imply the following. If for all polynomials $p(\cdot)$ and infinitely many n we have SQ-DIM$(\mathcal{F}_n, D_n) = d(n) \geq p(n)$ (that is, superpolynomial statistical query dimension), then $\mathcal{F}$ is not weakly learnable in the statistical query model with respect to D. On the other hand, if there exists a polynomial $p(\cdot)$ such that for all sufficiently large n, SQ-DIM$(\mathcal{F}_n, D_n) = d(n) \leq p(n)$ (that is, polynomial statistical query dimension), then there is a non-uniform (in n), polynomial time weak learning algorithm for $\mathcal{F}$ with respect to D in the statistical query model.

Note that Theorem 12, combined with the remarks above on the relationship between the VC and statistical query dimensions, implies that for any class of VC dimension d, there is a distribution on which $d^{1/3}/2$ statistical queries of tolerance $1/d^{1/3}$ must be made for PAC learning (this bound is incomparable to a similar lower bound given in the paper of Kearns [11]). However, as we have already noted, dramatically stronger lower bounds for statistical query learning may hold, even for natural input distributions. For instance, as promised, we have the following corollary.

[4] To see this, let D be uniform on a shattered set S of size $2^{\lfloor \log d \rfloor} \leq d$. Without loss of generality we may assume that $S = \{0,1\}^{\lfloor \log d \rfloor}$. Since S is shattered, the function class contains all possible boolean functions over S, and in particular the $\lfloor \log d \rfloor$ parity functions, which are pairwise uncorrelated.

Corollary 14 *There is a constant $c > 0$ such that $\Omega(n^{c \log n})$ statistical queries, each of tolerance $O(1/n^{c \log n})$, are required to weakly learn the classes of polynomial size DNF formulae and polynomial size decision trees with respect to the uniform distribution. Thus, these classes are not efficiently learnable in the statistical query model.*

Proof: Let D_n be the uniform distribution on $\{0,1\}^n$, and consider the class $\mathcal{F}_n$ of all parity functions χ_A over $\{0,1\}^n$ in which $|A| \leq \log n$ (thus, the parity function depends on at most $\log n$ of the input bits). The number of such functions is exactly $\binom{n}{\log n}$, and they are all pairwise uncorrelated with respect to D_n. Thus SQ-DIM$(\mathcal{F}_n, D_n) \geq \binom{n}{\log n}$. Furthermore, $\mathcal{F}_n$ is contained in both the class of polynomial size DNF formulae and polynomial size decision trees, since each function in $\mathcal{F}_n$ can be represented by its truth table restricted to the $\log n$ input bits on which the function depends. This truth table has size $O(2^{\log n}) = O(n)$ and can be represented as either a DNF of n terms or a decision tree of n leaves. The result follows by Theorem 12. □(Corollary 14)

We now turn to the proofs of the main theorems. Because the proof of Theorem 13 is significantly easier than the proof of Theorem 12, we give it first.

Proof of Theorem 13: For fixed $\mathcal{F}$ and D such that SQ-DIM$(\mathcal{F}, D) = d$, the nonuniform algorithm has "hard-wired" for each n a maximal set of functions $f_1, \ldots, f_d$ such that for all $i \neq j$,

$$|\mathbf{Pr}_D[f_i = f_j] - \mathbf{Pr}_D[f_i \neq f_j]| \leq \frac{1}{d^3}.$$

The algorithm makes d queries, each with tolerance $1/3d^3$. The ith query g_i is simply a request for the correlation of the target function with f_i, that is, $g_i(\vec{x}, \ell) = \ell \cdot f_i(\vec{x})$ (recall that by convention boolean functions assume values in $\{+1, -1\}$). By assumption, the set $\{f_1, \ldots, f_d\}$ is a maximal pairwise (nearly) uncorrelated set, so at least one query g_i will return a value of at least $1/d^3 - \tau \geq 2/3d^3$. Thus the algorithm has *found* an f_i such that

$$|\mathbf{Pr}_D[f_i = f] - \mathbf{Pr}_D[f_i \neq f]| \geq \frac{2}{3d^3} - \tau \geq \frac{1}{3d^3}$$

where f is the target function, and we can use f_i as our weak hypothesis. □(Theorem 13)

Proof of Theorem 12: In the following proof, it will be helpful to keep in mind that our eventual approach will be to perform a Fourier analysis not only of the functions in the target class $\mathcal{F}$, but also of the *query* function $g : \{0,1\}^n \times \{+1,-1\} \to \{+1,-1\}$. Recall that such a query is a request from the learner for an approximation to $\mathbf{E}_D[g(\vec{x}, f(\vec{x}))]$, where D is the target distribution and f is the target function.

In order to prove the theorem, we will need to use an extension of the Fourier theory to an arbitrary distribution; this extension has been examined in the computational learning theory literature before by Furst, Jackson and Smith [9]. Thus let D be an arbitrary probability distribution over $\{0,1\}^n$. Then for any two real-valued functions f and g over $\{0,1\}^n$, we can define the *inner product with respect to D* by

$$\langle f, g \rangle_D = \mathbf{E}_D[fg] = \sum_{\vec{x} \in \{0,1\}^n} D[\vec{x}] f(\vec{x}) g(\vec{x}).$$

It is easy to verify that $\langle , \rangle_D$ is in fact an inner product for the vector space of all real-valued functions over $\{0,1\}^n$, and we shall use this in our analysis. If, as usual, we regard the boolean functions $f_1, \ldots, f_d$ as being $\{+1,-1\}$-valued, then the assumption of the theorem gives that $|\langle f_i, f_j \rangle_D| \leq 1/d^3$ for all $i \neq j$. It is also easy to see that for any $\{+1,-1\}$-valued function f, $\langle f, f \rangle_D = 1$.

In the analysis to follow, we wish to use the given functions $f_1, \ldots, f_d$ as the beginnings of a basis for the vector space of all functions. To do this, we will need the following lemma.

Lemma 15 *For $d \geq 4$, the functions $f_1, \ldots, f_d$ are linearly independent.*

Proof: Without loss of generality, assume for contradiction that we could write $f_1 = \sum_{i \geq 2} \alpha_i f_i$ for some real coefficients $\alpha_2, \ldots, \alpha_d$. Then we have

$$\begin{aligned}
0 &= \mathbf{E}_D\left[(f_1 - \sum_{i\geq 2} \alpha_i f_i)^2\right] \\
&= \mathbf{E}_D[f_1^2] - 2\sum_{i \geq 2} \alpha_i \mathbf{E}_D[f_1 f_i] \\
&\qquad + \sum_{i,j \geq 2} \alpha_i \alpha_j \mathbf{E}_D[f_i f_j] \\
&= 1 - 2\sum_{i\geq 2} \alpha_i \mathbf{E}_D[f_1 f_i] + \sum_{i \geq 2} \alpha_i^2 \\
&\qquad + \sum_{i,j\geq 2, i\neq j} \alpha_i \alpha_j \mathbf{E}_D[f_i f_j]
\end{aligned}$$

where we have used that $\mathbf{E}_D[f_i^2] = 1$ for all i. Our goal is to reach a contradiction by showing that this final expression is strictly larger than 0. Let us define

$$\alpha_{\max} = \max\{|\alpha_i| : i \geq 2\} \geq 0$$

and use $\alpha_{\max}$ to simplify the expression above. Then

$$1 + \sum_{i \geq 2} \alpha_i^2 \geq 1 + \alpha_{\max}^2.$$

Also,

$$\left| 2\sum_{i \geq 2} \alpha_i \mathbf{E}_D[f_1 f_i] \right| \leq \frac{2\alpha_{\max}}{d^2}$$

since $\mathbf{E}_D[f_1 f_i] = \langle f_1, f_i \rangle_D \leq 1/d^3$ for all $i \neq 1$. Finally,

$$\left| \sum_{i,j \geq 2, i \neq j} \alpha_i \alpha_j \mathbf{E}_D[f_i f_j] \right| \leq \frac{\alpha_{\max}^2}{d}.$$

So we have:

$$\begin{aligned}
1 - 2\sum_{i\geq 2} &\alpha_i \mathbf{E}_D[f_1 f_i] + \sum_{i\geq 2} \alpha_i^2 \\
&+ \sum_{i,j \geq 2, i\neq j} \alpha_i \alpha_j \mathbf{E}_D[f_i f_j] \\
\geq\ & 1 + \alpha_{\max}^2 - \frac{2\alpha_{\max}}{d^2} - \frac{\alpha_{\max}^2}{d}.
\end{aligned}$$

Now if $\alpha_{\max} \leq 1$, then $2\alpha_{\max}/d^2 + \alpha_{\max}^2/d \leq 2/d^2 + 1/d < 1$ for $d \geq 3$. If $\alpha_{\max} > 1$, then $2\alpha_{\max}/d^2 + \alpha_{\max}^2/d \leq 3\alpha_{\max}^2/d < \alpha_{\max}^2$ for $d \geq 4$. In either case, we have $1 + \alpha_{\max}^2 - (2\alpha_{\max}/d^2 + \alpha_{\max}^2/d) > 0$, a contradiction. □(Lemma 15)

Before extending $f_1, \ldots, f_d$ to a complete basis, we argue that without loss of generality we can assume that the support of the input distribution D is all of $\{0,1\}^n$ [5]. The reason we may assume this is that if D does *not* have support $\{0,1\}^n$, we can instead carry out the ensuing analysis using a distribution D' that *does* have support $\{0,1\}^n$, and is obtained from D by taking an infinitesimally small amount of weight away from the support of D, and spreading this weight uniformly among the vectors not in the support of D. Then the functions $f_1, \ldots, f_m$ will still be approximately orthogonal, and it is not hard to prove that any statistical query lower bound we can prove for D' must also hold for D, since the learning algorithm cannot distinguish D and D' (details are omitted).

Now using the Gram-Schmidt process, which applies to any inner product space, we may extend $f_1, \ldots, f_d$ to obtain a basis $f_1, \ldots, f_d, f_{d+1}, \ldots, f_{2^n}$ for the vector space of all real functions over $\{0,1\}^n$ with the property that for any $i \geq d+1$ and any $j \neq i$, $\langle f_i, f_j \rangle_D = 0$, and for any i, $\langle f_i, f_i \rangle_D = 1$. Note that our basis may *not* be orthonormal due to the fact that for $i, j \leq d$, $\langle f_i, f_j \rangle_D$ may be as large as $1/d^3$. Also, note that we may assume there are 2^n basis functions: since D has support $\{0,1\}^n$, the 2^n delta functions on $\{0,1\}^n$ are orthogonal and non-zero with respect to D.

We now wish to extend $f_1, \ldots, f_{2^n}$ to a basis for the space of all real functions on $\{0,1\}^n \times \{+1,-1\}$. This obviously includes all such functions assuming values only in $\{0,1\}^n$, which is the space of possible statistical query functions. To accomplish this, it will be most convenient to use an inner product defined by a distribution $\tilde{D}$ on $\{0,1\}^n \times \{+1,-1\}$ that extends the distribution D, where $\tilde{D}$ is simply the product of D and the uniform distribution on $\{+1,-1\}$. To extend $f_1, \ldots, f_{2^n}$, we define for each $1 \leq i \leq 2^n$ the function $h_i(\vec{x}, y) = y \cdot f_i(\vec{x})$, where $\vec{x} \in \{0,1\}^n$ and $y \in \{+1,-1\}$. We also regard each of the original basis functions f_i as a function over $\{0,1\}^n \times \{+1,-1\}$, where $f_i(\vec{x}, y) = f_i(\vec{x})$. We now verify that $f_1, \ldots, f_{2^n}$ together with $h_1, \ldots, h_{2^n}$ is in fact a basis for all functions on $\{0,1\}^n \times \{+1,-1\}$ under the inner product $\langle , \rangle_{\tilde{D}}$. We have

$$\begin{aligned}
\langle h_i, f_j \rangle_{\tilde{D}} &= \frac{1}{2}\mathbf{E}_D[h_i(\vec{x}, 1) f_j(\vec{x})] \\
&\qquad + \frac{1}{2}\mathbf{E}_D[h_i(\vec{x}, -1) f_j(\vec{x})] \\
&= \frac{1}{2}\mathbf{E}_D[f_i(\vec{x}) f_j(\vec{x})] \\
&\qquad + \frac{1}{2}\mathbf{E}_D[-f_i(\vec{x}) f_j(\vec{x})] \\
&= 0
\end{aligned}$$

and

$$\begin{aligned}
\langle h_i, h_j \rangle_{\tilde{D}} &= \frac{1}{2}\mathbf{E}_D[h_i(\vec{x}, 1) h_j(\vec{x}, 1)] \\
&\qquad + \frac{1}{2}\mathbf{E}_D[h_i(\vec{x}, -1) h_j(\vec{x}, -1)] \\
&= \mathbf{E}_D[f_i f_j]
\end{aligned}$$

[5] If we regard D as a linear transformation of the vector space of all real functions over $\{0,1\}^n$ (that is, as a 2^n by 2^n matrix with nonzero entries only on the diagonal, corresponding to the probabilities assigned by D), this is simply saying that D has full rank.

and $\mathbf{E}_D[f_i f_j] = 0$ unless $i, j \leq d$, in which case it is bounded by $1/d^3$, or unless $i = j$, in which case it equals 1. So by the same argument as in Lemma 15 we have $2 \cdot 2^n$ independent functions, forming a basis for functions over $\{0,1\}^n \times \{+1,-1\}$.

Now let $g : \{0,1\}^n \times \{+1,-1\} \to \{+1,-1\}$ be any statistical query. We will soon perform a Fourier analysis of the expectation $\mathbf{E}_D[g(\vec{x}, f(\vec{x}))]$, which is the quantity that is approximated by the response of the query. Because we have a basis, we can write $g = \sum_{i \geq 1} \alpha_i f_i + \sum_{i \geq 1} \beta_i h_i$ for some real coefficients α_i and β_i. Note that it is not true that $\alpha_i = \langle g, f_i \rangle_{\tilde{D}}$ and $\beta_i = \langle g, h_i \rangle_{\tilde{D}}$ because we do not have an orthonormal basis. However, the following bound on the coefficients will serve our purposes.

Lemma 16 *If $g = \sum_{i \geq 1} \alpha_i f_i + \sum_{i \geq 1} \beta_i h_i$, where g and the f_i and h_i are as defined above, then $|\alpha_i|, |\beta_i| \leq 2$ for all i.*

Proof: Without loss of generality, let $\alpha_1 > 0$ be the largest coefficient. Since we have an inner product space, we can define the f_1-component of g by

$$\begin{aligned} \langle f_1, g \rangle_{\tilde{D}} f_1 &= \left(\alpha_1 + \sum_{i \geq 2} \alpha_i \langle f_1, f_i \rangle_D + \sum_{i \geq 1} \beta_i \langle f_1, h_i \rangle_{\tilde{D}} \right) f_1 \\ &= \left(\alpha_1 + \sum_{i \geq 2} \alpha_i \langle f_1, f_i \rangle_D \right) f_1 \end{aligned}$$

since $\langle f_1, h_i \rangle_{\tilde{D}} = 0$. (Note that if β_1 was the largest coefficient, we would instead take the h_1-component of g and proceed analogously.) Again due to the properties of an inner product, we must have

$$\begin{aligned} ||g|| &= \sqrt{\mathbf{E}_{\tilde{D}}[g^2]} \\ &\geq \left| \alpha_1 + \sum_{i \geq 2} \alpha_i \langle f_1, f_i \rangle_D \right|. \end{aligned}$$

But the summation inside the absolute value is at most α_1/d^2, so the absolute value is at least $\alpha_1 - \alpha_1/d^2 > \alpha_1/2$ for $d \geq 2$. Since $||g|| = 1$, the lemma follows. □(Lemma 16)

We are now finally in position to analyze the quantity of interest, the expected value of the query g. Let the target function be f_j for some $1 \leq j \leq d$; thus, we choose as the target one of the original nearly orthogonal functions in the target class $\mathcal{F}$. We may write:

$$\begin{aligned} \mathbf{E}_D[g(\vec{x}, f_j(\vec{x}))] &= \mathbf{E}_D \Bigg[\sum_{i \geq 1} \alpha_i f_i(\vec{x}) \\ &\qquad + \sum_{i \geq 1} \beta_i h_i(\vec{x}, f_j(\vec{x})) \Bigg] \\ &= \sum_{i \geq 1} \alpha_i \mathbf{E}_D[f_i] + \sum_{i \geq 1} \beta_i \mathbf{E}_D[f_i f_j] \\ &= C + \sum_{i \leq d} \beta_i \langle f_i, f_j \rangle_D \end{aligned}$$

where $C = \sum_{i \geq 1} \alpha_i \mathbf{E}_D[f_i]$ is a constant independent of the target function f_j, and we have used the fact that $\langle f_i, f_j \rangle_D = 0$ unless $i \leq d$. Now

$$\beta_j - \frac{2}{d^2} \leq \sum_{i \leq d} \beta_i \langle f_i, f_j \rangle_D \leq \beta_j + \frac{2}{d^2}$$

since $\langle f_j, f_j \rangle_D = 1$ and $\langle f_i, f_j \rangle_D \leq 1/d^3$ for $i \neq j$, and $|\beta_i| \leq 2$ for all i by Lemma 16. Thus, we see that the only contribution the target function makes to the expected value of the query is in determining the coefficient β_j, plus an $O(1/d^2)$ contribution. For the lower bound, the statistical query g will always be answered with the value C. We now analyze how many functions in $f_1, \ldots, f_d$ can be eliminated by this answer. For this, we need the following final lemma.

Lemma 17

$$\sum_{i \leq d} \beta_i^2 \leq 2.$$

Proof: Using Lemma 16 and the $1/d^3$ bounds on the inner products, it is easy to verify that $\mathbf{E}_{\tilde{D}}[g^2] = 1$ is bounded above and below by $\sum_{i \leq d} \alpha_i^2 + \sum_{i \leq d} \beta_i^2 \pm 16/d$. This implies $\sum_{i \leq d} \beta_i^2 \leq 1 - \sum_{i \leq d} \alpha_i^2 + 16/d \leq 2$ provided $d \geq 16$. □(Lemma 17)

Now if the query g is made with tolerance as large as $1/d^{1/3}$, then by the preceding arguments the function f_j is eliminated by the query response C only if β_j exceeds $1/d^{1/3}$. But by Lemma 17, if r is the number of functions f_j in $f_1, \ldots, f_d$ such that β_j exceeds $1/d^{1/3}$, then we must have $r(1/d^{1/3})^2 \leq 2$, or $r \leq 2d^{2/3}$. This shows that at least $d^{1/3}/2$ queries of allowed approximation error bounded above by $1/d^{1/3}$ are required in order to eliminate all the functions in $f_1, \ldots, f_d$ that are not the target function. If there are even two functions remaining, by choosing adversarially between the remaining functions we may force the error of the learning algorithm's hypothesis to be $1/2 - 1/d^3$ with significant probability. □(Theorem 12)

Note that in the above proof, even if the learning algorithm is randomized, if it makes only $d^{1/3-\epsilon}$ queries for some constant $\epsilon > 0$, it will eliminate only a small fraction of $f_1, \ldots, f_d$. So if the adversary picks f_j at random from $f_1, \ldots, f_d$, with high probability it can answer as above for each query and so again with high probability the algorithm's error will be close to $1/2$.

It is also instructive to note that if the tolerance $\tau = 0$, then over the uniform distribution the statistical query model allows one to make membership queries (one can ask whether the probability of a specific labeled example is nonzero). So the algorithmic results in the previous sections prove that such a lower bound cannot hold when 0-tolerance queries may be made.

6 Acknowledgments

This manuscript is submitted for publication with the understanding that the U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notation thereon.

References

[1] Howard Aizenstein, Lisa Hellerstein, and Leonard Pitt. Read-thrice DNF is hard to learn with membership and

equivalence queries. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 523–532, 1992.

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

[3] Howard Aizenstein and Leonard Pitt. Exact learning of read-k disjoint DNF and not-so-disjoint DNF. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 71–76, 1992.

[4] Dana Angluin, Michael Frazier, and Leonard Pitt. Learning conjunctions of Horn clauses. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 186–192, 1990.

[5] Avrim Blum and Steven Rudich. Fast learning of k-term DNF formulas with queries. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 382–389, 1992.

[6] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warumth. Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*, 36(4):929–965, October 1989.

[7] Bruck, J. Harmonic Analysis of Polynomial Threshold Functions. *SIAM Journal of Discrete Mathematics*, 5, pp. 168–177, 1990.

[8] Nader H. Bshouty. Exact learning via the monotone theory. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 302–311, 1993.

[9] Merrick L. Furst, Jeffrey C. Jackson, and Sean W. Smith. Improved learning of AC^0 functions. In *Fourth Annual Workshop on Computational Learning Theory*, pages 317–325, 1991.

[10] Thomas R. Hancock. Learning 2μDNF formulas and $k\mu$ decision trees. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 199–209, 1991.

[11] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 392–401, 1993.

[12] Michael Kearns, Ming Li, Lenny Pitt and Leslie G. Valiant. On the learnability of Boolean formulae. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 285–295, 1987.

[13] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. In *Fifth Annual Workshop on Computational Learning Theory*, pages 341–352, 1992.

[14] Roni Khardon. On using the Fourier transform to learn disjoint DNF. Unpublished manuscript, September 1993.

[15] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 372–381, 1993.

[16] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing* 22(6):1331–1348, December 1993. (Also in *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 455–464, 1991.)

[17] Eyal Kushilevitz and Dan Roth. On learning visual concepts and DNF formulae. In *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, pages 317–326, 1993.

[18] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM* 40(3):607–620, July 1993. (Also in *30th Annual Symposium on Foundations of Computer Science*, pages 574–579, 1989.)

[19] Richard Lipton. Personal communication.

[20] Yishay Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. In *Fifth Annual Workshop on Computational Learning Theory*, pages 53–61, 1992.

[21] Lenny Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4): 965–984, October 1988.

[22] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106.

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