Derandomized Learning of Boolean Functions

Meera Sitharam[1] and Timothy Straney[2]

[1] Department of Math and CS, Kent State University, Kent OH 44240, sitharam@mcs.kent.edu
[2] Department of Math and CS, Kent State University, Kent OH 44240, tstraney@mcs.kent.edu

Abstract. We define a new model of learning called the *Always Approximately Correct* or AAC model. In this model the learner does not have random bits at its disposal, and instead learns by making the usual membership queries from a deterministic "training set." This model is an extension of Angluin's Query model of exact learning with membership queries alone.

We discuss crucial issues and questions that arise when this model is used. One such question is whether a *uniform* training set is available for learning any concept in the concept class. This issue seems not to have been studied in the context of Angluin's Query model. Another question is whether the training set can be *found* quickly if partial information about the function is given to the learner (in addition to the answers to membership queries); for example information about a subclass in which the concept belongs. We formalize the latter scenario by introducing the notion of "subclass queries."

Using this new model of learning, we prove three learnability results for classes of Boolean functions that are approximable (with respect to various norms) by linear combinations of a set of few *Parity* functions. We compare and contrast these results with several existing results for similar classes in the PAC model of learning with and without membership queries – these classes have not been previously emphasized under Angluin's Query model for exact learning.

Moreover, we point out the significance - in various contexts - of the classes of Boolean functions being learnt, for example in the context of probabilistic communication complexity.

1 Introduction

Learning algorithms for several classes of Boolean functions have exploited the (Fourier) spectral properties of functions in the class. These include the learning algorithms for AC^0 functions in [21], [12], [24], for decision trees in [20], for DNF formulae in [19], and for monotone Boolean functions in [8].

These algorithms, in effect, deal with classes of Boolean functions f that are approximable by some linear combination g of few *Parity* (or Fourier basis) functions, with respect to a chosen norm, and the algorithms obtain such an approximation g as a hypothesis. The *Parity* functions can also be viewed as a

monomial basis for the space of functions over the cube, if the cube is taken as $\{-1,1\}^n$.

In some cases, the set of *Parity* functions that define the approximating class is fixed (and known to the learner), as in [21], [24], and [12], and in others, its size is fixed (and known to the learner), but the set itself is variable and left for the learner to decipher, as in [20] and [19].

For some of these algorithms, the bound on the probability ϵ that the hypothesis is erroneous on a random input is determined by the distance of the best approximation g to the function f from the given class, or is otherwise determined by some characteristics of the class being learnt. I.e, the hypothesis and learning algorithm are "weak." This includes the algorithms for learning functions approximable in the 2-norm by polynomially many *Parity* functions in [20]. In other cases, ϵ can be chosen freely and the hypothesis class can be appropriately enlarged. The running time of such "strong" learning algorithms typically depends (polynomially) on $1/\epsilon$. This includes the algorithm in [20] for learning functions whose Fourier expansion has a small L_1 norm, which applies (as observed in [11]) to learning functions whose sign can be expressed as a linear combination of polynomially many *Parity* functions, i.e, the class PT^1. This also includes the algorithms in [19] for learning DNF functions, and functions that are approximable in sign by a small linear combination of polynomially many *Parity* functions; and algorithms for learning AC^0 functions in [21] and [24].

In a few cases, the function f is exactly reproduced (i.e, with error $\epsilon = 0$), with high probability in polynomial time as in the case of a learning algorithm of [20] for learning decision trees.

All of the above algorithms (except [24]) use the PAC model of learning with respect to the uniform distribution, and certain other distributions; sometimes the algorithms use learner-specified membership queries, in addition. The setting is therefore *apriori assumed* to be probabilistic: the queries are chosen randomly with respect to some distribution D and with high probability the algorithm outputs a hypothesis with small probability of error on an input chosen randomly with respect to the same distribution D. In these models, even "exact" learning algorithms such as the algorithm in [20], rely on random sampling in addition to membership queries and output the (correct) hypothesis only with high probability.

In [24], in contrast, it was shown that in some cases the PAC model is unnecessary, and, in a sense misleading. One result of this paper was a derandomization of the PAC learning algorithm of [21] for AC^0 functions.

The algorithm in [21] relies on their elegant result (based on Hastad's switching lemma [16]) that AC^0 functions are approximable in the 2-norm, within any chosen ϵ, by a linear combination of the *Parity* functions in a certain class. This class consists of *small weight Parity* functions, i.e, *Parity* functions which evaluate the parity of a set of at most polylogarithmically many bits. If *Parity* functions are viewed as monomials, then these linear combinations are small degree polynomials. In [21], a strong learning algorithm is given which runs in moderately superpolynomial time and learns AC^0 functions using membership

queries, which are chosen randomly with respect to the uniform distribution. The strong hypothesis produced (with high probability) is the sign (or Booleanization) of a linear combination of *small weight Parity* functions. The bound ϵ on the probability that the hypothesis is erroneous on a random input can be chosen as desired. The running time of the algorithm grows as a moderate superpolynomial in $O(1/\epsilon)$.

The result in [24] shows that for any desired error probability ϵ, there is a *single, deterministic* set of membership queries or a *training set* that applies to *all* AC^0 functions (computable within a circuit size and depth bound) and achieves the same purpose in the same time. In fact, it was shown that any set of inputs that appear random and "fool" AC^0 functions, can serve this purpose as a uniform training set.

The algorithm in [24] almost, (but not quite), fits the framework of Angluin's well-studied, deterministic Query model of exact learning [2]. In this model as well, the queries are deterministically chosen by the learner, but the hypothesis produced always, *exactly* matches the concept being learnt. The algorithm in [24], however, allows a hypothesis error, and allowing this error is *crucial* to the existence of the algorithm.

Several classes of Boolean functions have been studied, and nice results have been obtained under Angluin's Query model, such as [3], [4] [7]. However, many of these algorithms use other types of queries in addition to membership queries in order to obtain exact hypotheses. Moreover, notice that none of the classes studied under the Query model is defined based on approximability from few *Parity* or Fourier basis functions, or sparse multilinear real polynomials over the cube domain, i.e, $\{-1,1\}^n$. These classes of functions, are however, the primary interest in this paper (and have been studied under the PAC model, as described earlier). These classes have been dealt with in the query model, only when the domain is $\mathbb{Z}^n$, in [6], and [25] (which uses counterexamples in addition to membership queries); and for the case of polynomials over finite fields, in [9]. For the cube domain, Furthermore, while the algorithms studied under Angluin's Query model make a deterministic (training) set of queries, the natural issue - of whether this training set is *uniform* over the concept class - has not been emphasized or well-investigated.

Model and Relevant Issues. The result in [24], in effect, used a stronger model of learning than the PAC model with membership queries, and a more general model than Angluin's Query model. The new model defined here is called AAC or Always Approximately Correct. In this model, the learner needs no random bits, and produces a hypothesis that is *always* approximately correct to within some fixed, reasonable ϵ that depends on the class being learnt. The hypothesis error is measured with respect to the uniform distribution on the inputs. We restrict the definition to the learning of Boolean functions with respect to the uniform distribution. The definition can be generalized to other concepts and distributions; it can be made distribution independent; and a strong learning version can be defined by allowing a free choice of ϵ which influences the

running time, but we avoid these generalizations in this paper since we do not require them. Crucial questions that arise in this model of learning are:

- Do deterministic training sets exist for the functions in the given class?
- Do these sets have small size (since this affects the running time)?
- Is the set a *single, uniform* set independent of the particular function being learnt, and depending only on the class?
- What characterizes the structure of training sets? In particular, what is the relationship of a training set to sets of pseudorandom strings that appear random and fool functions in the class?
- How much time does it take to *find* a training set, given some finite, partial description of the function to be learnt, in addition to answers to the usual membership queries?

Scope, Results and Significance. In this paper, we limit our scope to functions over the vertices of the cube $\{0,1\}^n$ or $\{-1,1\}^n$, not necessarily Boolean valued, which are approximable by a linear combination of a set of *Parity* functions. The set is either fixed, and known to the learner prior to the learning phase, or it is variable, but *input to the learner* during the learning phase along with the membership query information about the function being learnt. In the latter case, the running time of the algorithm *includes* the time required by the learner to *find* the training set to use during the learning phase.

More specifically, we prove 3 results. Below we discuss the results and their significance.

(1) The first result concerns functions f over the cube that are exactly expressible as a linear combination of a set of m *Parity* functions. If the set Q of *Parity* functions is variable and unknown to the learner, the problem reduces to a blackbox-interpolation question by m-sparse, real-valued polynomials. This has been dealt with, e.g, in [6], and [25] (which uses counterexamples in addition to membership queries) when the domain is $\mathbb{Z}^n$, and for the case of polynomials over finite fields, in [9]. For the cube domain, the learning algorithm of [20] gives an exact reproduction of the function f, but since it works in the PAC model, it uses randomly chosen membership queries, and the exact reproduction is output with high probability, but not always.

On the other hand, if the set Q of parity functions is fixed and known to the learner, reproducing the function f exactly - i,e, determining the (Fourier) coefficients of the linear combination that gives f - is a simple black-box interpolation question which can be solved deterministically, with no randomness required of the learner. The (uniform) training set for all functions approximable from Q is hard-coded into the learner, the learner poses one point-evaluation or membership query to the black-box/teacher for each element of the training set; and solves a Vandermonde-type interpolation system to obtain the coefficients, and therefore f. The uniform training set for functions in Q is chosen *apriori* to ensure that the interpolation system for Q is non-singular and can be solved.

In this paper, we consider the case where the set Q is neither fixed nor known to the learner. It is variable, but its elements are *input* to the learner during the learning phase as an answer to a *subclass query* from the learner, distinct from a membership query. The learner is required to *find* the uniform training set for each specific Q which would ensure that the Vandermonde interpolation system is nonsingular. The time to find the training set is included in the running time of the learning algorithm.

We obtain a learning algorithm in the AAC model with subclass queries, that exactly reproduces functions f that are linear combinations of a set Q of parity functions where the set is variable, but is a subspace of $\mathbb{F}_2^n$. The algorithm runs in time $O(|Q|^4)$. The algorithm utilizes a clean description of the *structure* of training sets (called *subspace-like*) that are appropriate for subspaces Q. The description of the structure of good training sets extends to the case when Q itself is only subspace-like although there is no obvious corresponding extension of the learning algorithm.

(2) The second result considers the efficacy of the AAC model for learning Boolean functions that are only *approximable* in the 2 norm by linear combinations of *Parity* functions in a set Q. Here we simply observe that there is *no* uniform training set that applies to all functions in this class, unlike the situation in the first result. Note that this does not, however, preclude the existence of an AAC algorithm for learning functions in this class, since the training sets could be constructed dynamically depending on the specific function being learnt, using the membership queries.

(3) The third result considers the efficacy of the AAC model for learning $\{+1, -1\}$ valued Boolean functions that are closely *approximable* to within some $\epsilon < 1/2$, typically, in our cases, $\epsilon < 1/(4|Q|)$ is meaningful. The approximation is in the ∞ norm, by linear combinations of *Parity* functions in a set Q. (Notice that this class is a subclass of $\widehat{PT^1}$ functions, if $|Q|$ is polynomially bounded).

We consider two cases: when the set Q is fixed and known to the learner, and when it is variable, but input to the learner via subclass queries. In the former case, we consider general sets Q, as well as sets Q that are subspace-like, but in the latter case, we only consider sets Q that are subspaces.

In both cases, we prove the existence of a uniform training set for all functions in the class, and for the more specific sets Q, we utilize the structure of the training sets developed for the first result described above. In addition, we point out the relationship between the training sets and sets of pseudorandom strings for this class.

For Q fixed and subspace-like, we give an algorithm that runs in time $O(|Q|^2)$ to produce a hypothesis that errs on at most $O(|Q|\epsilon^2)$ fraction of the inputs i.e, errs on a random input from the uniform distribution with probability at most $O(|Q|\epsilon^2)$.

For Q that are variable subspaces, we adapt the algorithm obtained in the first result that uses subclass queries (and obtains the description of Q as input),

finds the uniform training set, makes the corresponding membership queries and outputs the hypothesis, all in time $O(|Q|^4)$. The hypothesis error is bounded by by $O(|Q|\epsilon^2)$.

Classes of Boolean functions with approximations in the ∞ norm have been studied often in the context of threshold circuits (see [26]), since obtaining the sign of the Boolean function is the same as approximating it in the ∞ norm. However, we study classes of Boolean functions with *close* approximations. Such classes have been studied by [22], [23], and [14] in the context of analytic and combinatorial properties of Boolean functions. In both of these papers, the set Q is fixed to be the class of small-weight *Parity* functions or low degree monomials.

Close ∞ norm approximation - from basis functions that are characteristic functions of cross-product sets or combinatorial rectangles, rather than directly from *Parity* functions - arises naturally in the context of probabilistic communication complexity, and is hidden in all proofs where probabilistic communication complexity is used as a tool for proving threshold circuit lower bounds, for example in [18] and [13]. The use of this notion of approximation in this context is clarified in [11] and relies on the following

Fact If the $(1-\epsilon)$-error probabilistic communication complexity of a Boolean function f is at most $\log m$, then there is an approximation g with the same sign as f, of the form $g = \sum_{i \leq m^2} a_i r_i$, where $\sum_{i \leq m} |a_i| \leq 1$, the r_i are characteristic functions of cross-product sets or combinatorial rectangles, and $|g(x)| \geq \epsilon/m$ everywhere. In other words f can be well-approximated as a linear combination of at most m^2 combinatorial rectangles.

Although we deal with *Parity* basis rather than a combinatorial rectangle basis, combinatorial rectangles decompose as special linear combinations of *Parity* functions, i.e, their Fourier spectra have specific properties, see for example [15]. For this reason, our results are potentially useful in obtaining learning algorithms for functions that have certain types of probabilistic communication protocols.

Organization. Section 2 gives basic conventions and background on Fourier transforms and Hadamard matrices and precisely defines the AAC learning model for classes of Boolean functions, as well as the concept of subclass queries. Sections 3, 4 and 5 deal with the first, second and third result described above. Section 6 discusses conjectures and open problems.

2 Background and Preliminaries

The following terminology and conventions will be used throughout this paper. $\mathbb{F}_2^n$ is a finite vector space over $\mathbb{F}_2$ consisting of all n-tuples with entries from $\mathbb{F}_2$ and inner product $\langle , \rangle$. If $x \in \mathbb{F}_2^n$, define $P_x = \{y \in \mathbb{F}_2^n | (-1)^{\langle x,y \rangle} = 1\}$. In general for any subset Q of $\mathbb{F}_2^n$, the set $\bar{Q}$ is the complement of Q in $\mathbb{F}_2^n$, and

$Q^\perp$ is defined as the set of all vectors y that are orthogonal to every vector in Q, i.e, the vectors that belong in P_x for every $x \in Q$.

If Q is a subspace of $\mathbb{F}_2^n$, then so is $Q^\perp$ and both are also subgroups of $\mathbb{F}_2^n$. As such, both group theoretic and vector space properties apply. In this paper we will often apply group theoretic properties to Q, while referring to it in the larger sense as a subspace of $\mathbb{F}_2^n$.

If $x, y \in \mathbb{F}_2^n$, define the *Parity* function $\chi_x : \mathbb{F}_2^n \to \{-1, 1\}$ by $\chi_x(y) = (-1)^{\langle x,y \rangle}$. Notice that if $\mathbb{F}_2^n$ is viewed as $\{-1,1\}^n$, then $\chi_x(y)$ is the monomial $\prod_{i:x^i=-1} y^i$, where x^i and y^i here denote the i^{th} entries in x and y respectively. Often, we identify a subset Q of $\mathbb{F}_2^n$ with the set of *Parity* functions $\{\chi_x : x \in Q\}$.

The set of real valued functions on $\mathbb{F}_2^n$ is a vector space of dimension 2^n with basis $\{\chi_x | x \in \mathbb{F}_2^n\}$. (Unless otherwise specified, Boolean functions are assumed to be $\{-1,1\}$-valued.) If f and g belong to this space, define an inner product by $\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)g(x)$. Note that $\{\chi_x | x \in \mathbb{F}_2^n\}$ is an orthonormal basis under this inner product, since $\langle \chi_x, \chi_y \rangle = 0$, if $x \neq y$ and 1 otherwise. The norm $\|f\|_2 = \sqrt{\langle f, f \rangle}$; $||f||_\infty$ is simply $\max_{x \in \mathbb{F}_2^n} |f(x)|$, and $||f||_1$ is $\sum_{x \in \mathbb{F}_2^n} |f(x)|$. Unless otherwise specified, "polynomially bounded" means "polynomially bounded in n."

If $f : \mathbb{F}_2^n \to \mathbb{R}$, then its Fourier transform is a function $\hat{f} : \mathbb{F}_2^n \to \mathbb{R}$ defined by $\hat{f}(y) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)\chi_x(y) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)\chi_y(x) = \langle f, \chi_y \rangle$. Furthermore, for every $x \in \mathbb{F}_2^n$, $f(x) = \sum_{y \in \mathbb{F}_2^n} \hat{f}(y)\chi_x(y) = \sum_{y \in \mathbb{F}_2^n} \hat{f}(y)\chi_y(x)$. The support of $f : \mathbb{F}_2^n \to \mathbb{R}$ is $\{x \in \mathbb{F}_2^n | f(x) \neq 0\}$ and is denoted spt f. If $f : \mathbb{F}_2^n \to \mathbb{R}$, then by Parseval's identity $\frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)^2 = \sum_{y \in \mathbb{F}_2^n} \hat{f}(y)^2$, which follows from the orthonormality of $\{\chi_x | x \in \mathbb{F}_2^n\}$

The Hadamard matrix is used extensively. The Hadamard matrix H_{2^n} is a $\{+1, -1\}$ matrix consisting of rows x and columns y labeled by elements of $\mathbb{F}_2^n$ with the x, y^{th} entry being $\chi_x(y)$. The Hadamard matrix is an orthogonal matrix. Furthermore the row and column labels used to form H_{2^n} can be ordered in such a way that H_{2^n} is a symmetric matrix, in which case, $H_{2^n} = 2^n H_{2^n}^{-1} = H_{2^n}^T$.

The Classes and the Model. The following classes of Boolean functions are considered.

Definition 1. Let $n \in \mathbb{N}$, $Q \subseteq \mathbb{F}_2^n$ and ϵ be a nonnegative constant. Then

$C_0^Q = \{f : \mathbb{F}_2^n \to \mathbb{R} | \text{spt } \hat{f} \subseteq Q\}$

$C_\epsilon^{Q,2} = \{f : \mathbb{F}_2^n \to \{0,1\} | \exists g : \mathbb{F}_2^n \to \mathbb{R} \text{ such that spt } \hat{g} \subseteq Q \text{ and } \|f - g\|_2^2 \leq \epsilon\}$

$C_\epsilon^{Q,\infty} = \{f : \mathbb{F}_2^n \to \{0,1\} | \exists g : \mathbb{F}_2^n \to \mathbb{R} \text{ such that spt } \hat{g} \subseteq Q \text{ and } \|f - g\|_\infty \leq \epsilon\}$

In general, $C_0^\Pi = \cup_{Q \in \Pi} C_0^Q$ over all sets Q belonging to some class Π. and $C_\epsilon^{\Pi,\infty} = \cup_{Q \in \Pi} C_\epsilon^{Q,\infty}$.

In dealing with the last class above, we use the following duality theorem for Boolean functions. See [26] and [10].

Theorem 2. *If $f \in C_\epsilon^{Q,\infty}$, then for all $s \in C_0^{\bar{Q}}$, with $||s||_1 \leq 1$, $2^n|\langle f, s\rangle| = |\sum_x f(x)s(x)| \leq \epsilon$. Here $\bar{Q}$ is the complement of Q in $\mathbb{F}_2^n$.*

Next we define the AAC model for weakly learning Boolean functions with respect to membership queries and the uniform distribution. As pointed out in the Introduction, this can be easily extended, if necessary, to other concepts and distributions; distribution-independent and strong learning versions are also natural extensions. For example, the algorithm in [24] uses a strong learning version of the model, since the algorithm works (appropriately fast) for any desired error bound that is input.

Definition 3. A class C of Boolean functions f over $\mathbb{F}_2^n$ is *AAC learnable* if there is a deterministic learning algorithm that uses membership queries to f from a deterministic *training set* and outputs a hypothesis h such that h differs from f on at most an $\epsilon_C < 1/2$ fraction of the 2^n inputs, where ϵ_C is determined by the characteristics of the class C. The algorithm should run in time bounded by a polynomial in $|f|$, which is the size of some finite representation of f (usually related to the hypothesis class). If the training set used by the algorithm is the same for all functions in the class C, it is called a *uniform* training set for C, and is assumed to be "hard-coded" into the learning algorithm.

Remark. See Section 1 for a description of the relationship of this model to Angluin's Query model, and previous results, and issues that have been studied in the context of that model.

We will use the following simple folklore theorem relating the distance of a function to its best approximation and the error of its best hypothesis.

Theorem 4. *If a Boolean function $f \in C_\epsilon^{Q,2}$ has an approximation $g \in C_0^Q$ such that $||f - g||_2^2 \leq \epsilon$, ($\epsilon < 1/2$ to be meaningful) then g is a good hypothesis for f whose sign differs from f on at most an ϵ fraction of $\mathbb{F}_2^n$. The best approximation g is typically taken as the projection of f on the Fourier basis functions given by Q, i.e, the function that satisfies $\hat{g}(x) = \hat{f}(x)$ when $x \in Q$ and $\hat{g}(x) = 0$ when $x \notin Q$.*

Finally we define the concept of learning with subclass-queries.

Definition 5. The class C_0^{Π} (or $C_\epsilon^{\Pi,\infty}$) is said to be AAC learnable with *subclass queries* if there is a deterministic AAC learning algorithm that receives answers to membership queries about the function f to be learnt and in addition receives the set $Q \in \Pi$ such that f belongs to the subclass C_0^Q of C_0^{Π} (or the subclass $C_\epsilon^{Q,\infty}$ of $C_\epsilon^{\Pi,\infty}$). The running time of the algorithm includes the time it takes to *construct* the training set.

3 The Class C_0^Q

We begin this section by recalling that if $Q = \{q_1, \ldots, q_m\}$ is any subset of $\mathbb{F}_2^n$ and $f \in C_0^Q$, then $f = \hat{f}(q_1)\chi_{q_1} + \cdots + \hat{f}(q_m)\chi_{q_m}$. Thus in order to learn f *exactly* in the AAC model one may select m elements from $\mathbb{F}_2^n$, say $x_1, \ldots, x_m$, such that the set of vectors $\{(\chi_{q_1}(x_1), \ldots, \chi_{q_m}(x_1)), \ldots, (\chi_{q_1}(x_m), \ldots, \chi_{q_m}(x_m))\}$ is linearly independent. After sampling f on each of $x_1, \ldots, x_m$, one formulates and solves the system:

$$\begin{bmatrix} \chi_{q_1}(x_1) & \cdots & \chi_{q_m}(x_1) \\ \vdots & & \vdots \\ \chi_{q_1}(x_m) & \cdots & \chi_{q_m}(x_m) \end{bmatrix} \begin{bmatrix} \hat{f}(q_1) \\ \vdots \\ \hat{f}(q_m) \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}$$

to obtain the Fourier coefficients of f, thus learning f exactly. If the set Q is fixed and known to the learner prior to the learning phase, then it is assumed the set, $\{x_1, \ldots, x_m\}$, that makes $\{(\chi_{q_i}(x_1), \ldots, \chi_{q_m}(x_1)), \ldots, (\chi_{q_1}(x_m), \ldots, \chi_{q_m}(x_m))\}$ linearly independent is also known and thus we have a learning algorithm for f that runs in time polynomial in $|Q|$. On the other hand if the set Q is variable and is given as input to the learner, then finding an appropriate training set, $\{x_1, \ldots, x_m\}$, could require exhaustive search.
In the following we show that for certain subsets Q of $\mathbb{F}_2^n$, there are training sets Q' which can be *found* in time polynomial in $|Q|$.

Definition 6. Let S be a subspace of $\mathbb{F}_2^n$ of order m. A subset $S' = \{x_1, \ldots, x_m\}$ of $\mathbb{F}_2^n$ is a *subspace-like* set derived from S if $x_1 \in S^\perp$, $x_i \notin S^\perp$ for $i \in \{2, 3, \ldots, m\}$, and $x_i + x_j \notin S^\perp$, $\forall i, j$, $i \neq j$.
Note that every subspace S' is a subspace-like set derived from some subspace S. S could be taken as S' itself, provided $S' \cap S'^\perp = \{0\}$.

Proposition 7. *Let $Q = \{q_1, \ldots, q_m\}$ be a subspace of $\mathbb{F}_2^n$ with basis $\{q_1, \ldots, q_{\log m}\}$. Let x_1 be an arbitrary element of $Q^\perp$. Let $\{S_2, S_3, \ldots, S_m\}$ be the set of nonempty subsets of $\{1, 2, \ldots, \log m\}$. For each S_i, let $x_i \in \mathbb{F}_2^n$ be a solution to the system:*

$$k \in S_i, \langle q_k, x_i \rangle = 1$$
$$k \notin S_i, \langle q_k, x_i \rangle = 0$$

Then $Q' = \{x_1, \ldots, x_m\}$ is a subspace-like set derived from Q. Moreover, given Q and a basis for Q, Q' can be found in time at most $O(|Q|^4)$.

Proof. Begin by noting that since $\{q_1, \ldots, q_{\log m}\}$ is linearly independent, the system in the proposition has a unique solution for each i. By hypothesis $x_1 \in Q^\perp$. Since S_i is nonempty for each $i \in \{2, 3, \ldots, m\}$, $\langle q_i, x_i \rangle = 1$ and consequently, $x_i \notin Q^\perp$ for each $i \in \{2, \ldots, m\}$.

Claim: If $i, j \in \{1, \ldots, m\}$ and $i \neq j$, then $x_i + x_j \notin Q^\perp$.
Pf. Suppose $i, j \in \{1, 2, \ldots, m\}$ and $i \neq j$.

CASE 1: $i = 1$
Since $i = 1$, $x_i \in Q^\perp$. Since $j \neq 1$, $x_j \notin Q^\perp$. Thus there exists $q \in Q$ such that $\langle q, x_i + x_j \rangle = \langle q, x_i \rangle + \langle q, x_j \rangle = 1$. Hence $x_i + x_j \notin Q^\perp$.

CASE 2: $i \neq 1 \neq j$
In this case either $S_i \setminus S_j \neq \emptyset$ or $S_j \setminus S_i \neq \emptyset$. With out loss of generality assume $S_i \setminus S_j \neq \emptyset$. Let $k \in S_i \setminus S_j$. Then $\langle q_k, x_i + x_j \rangle = \langle q_k, x_i \rangle + \langle q_k, x_j \rangle = 1$. Thus $x_i + x_j \notin Q^\perp$.
It follows that $Q' = \{x_1, \ldots, x_m\}$ is a subspace-like set derived from Q.

Finally note that to find each element of Q', the system in the proposition needs to be solved, which can be done in time at most $O(|Q|^3)$. Thus Q' can be found in time $O(|Q|^4)$. □

Proposition 8. *If Q is a subspace of $\mathbb{F}_2^n$ and $x \in \mathbb{F}_2^n \setminus Q^\perp$, then $|\{q \in Q : \chi_q(x) = 1\}| = |\{q \in Q : \chi_q(x) = -1\}|$.*

Proof. Note that if $Q =< 0^n >$, then the proposition is vacuously true. So we assume $|Q| \geq 2$. Let $Q = \{q_1, \ldots, q_{2^m}\}$ be a subspace of $\mathbb{F}_2^n$ of size 2^m, where $1 \leq m \leq n$. Let $x \in \mathbb{F}_2^n \setminus Q^\perp$. Then there exists $y \in Q$ such that $\chi_y(x) = -1$. Consider the map $\phi : Q \to Q$ given by $\phi(q_i) = y + q_i$, $\forall i \in \{1, \ldots, 2^m\} = I$.

Claim A: ϕ is a bijection
Pf. Let $q_i, q_j \in Q$ and suppose $\phi(q_i) = \phi(q_j)$. Then $y + q_i = y + q_j$ and $q_i = q_j$. Thus ϕ is injective.
Let $q_i \in Q$. Then $q_i - y \in Q$ and $\phi(q_i - y) = y + (q_i - y) = q_i$. Thus ϕ is surjective.

Claim B: For every $i \in I$, $\chi_{\phi(q_i)}(x) = -1$ if and only if $\chi_{q_i}(x) = 1$
Pf. Let $i \in I$. Then $\chi_{\phi(q_i)}(x) = -1$ if and only if $\chi_{y+q_i}(x) = -1$ if and only if $\chi_y(x) \cdot \chi_{q_i}(x) = -1$ if and only if $-1 \cdot \chi_{q_i}(x) = -1$ if and only if $\chi_{q_i}(x) = 1$.

Claim C: For every $i \in I$, $\chi_{\phi(q_i)}(x) = 1$ if and only if $\chi_{q_i}(x) = -1$
Pf. Since $\chi_{q_i}(x) \in \{-1, 1\}$, $\forall i \in I$, this claim follows immediately from Claim B.

Claim D: $|\{q \in Q : \chi_q(x) = 1\}| = |\{q \in Q : \chi_q(x) = -1\}|$
Pf. Let $J = \{i \in I : q_i \in Q \text{ and } \chi_{q_i}(x) = 1\}$. Assume $|J| > \frac{|I|}{2}$. Then $\chi_{q_i}(x) = -1$ for less than half the values $i \in I$. By Claim B $\chi_{\phi(q_i)}(x) = -1$, $\forall i \in J$. Since ϕ is a bijection, this means $\chi_{q_i}(x) = -1$ for more than half the values $i \in I$. Contradiction. Therefore $|J| \leq \frac{|I|}{2}$.

Let $K = \{i \in I : q_i \in Q \text{ and } \chi_{q_i}(x) = -1\}$. Assume $|K| > \frac{|I|}{2}$. Then $\chi_{q_i}(x) = -1$ for more than half the values $i \in I$. By Claim C $\chi_{\phi(q_i)}(x) = 1$, $\forall i \in K$. Since ϕ is a bijection, this means $\chi_{q_i}(x) = 1$ for more than half the values $i \in I$. Contradiction. Therefore $|K| \leq \frac{|I|}{2}$.
Since $\chi_{q_i}(x) \in \{-1, 1\}$, $\forall i \in I$ and $|J| \leq \frac{|I|}{2}$ and $|K| \leq \frac{|I|}{2}$, it follows that $|\{q \in Q : \chi_q(x) = 1\}| = |J| = \frac{|I|}{2} = |K| = |\{q \in Q : \chi_q(x) = -1\}|$. □

Theorem 9. *Let $S = \{x_1, \ldots, x_m\}$ be a subspace-like set derived from a subspace $Q = \{q_1, \ldots, q_m\} \subseteq \mathbb{F}_2^n$. If $H_{S,Q} = [h_{ij}]$ is the $m \times m$ matrix given by $h_{ij} = \chi_{q_j}(x_i)$, then $H_{S,Q}$ is a Hadamard matrix.*

Proof. Let $i, j \in \{1, 2, \ldots, m\}$ with $i \neq j$. Since $S = \{x_1, \ldots, x_m\}$ is a subspace-like set derived from Q, $x_i + x_j \notin Q^{\perp}$. Thus $|\{q \in Q : \chi_q(x_i) = \chi_q(x_j)\}| = |\{q \in Q : \chi_q(x_i + x_j) = 1\}| = |\{q \in Q : \chi_q(x_i + x_j) = -1\}| = |\{q \in Q : \chi_q(x_i) \neq \chi_q(x_j)\}|$, by Prop. 8. Therefore $\langle(\chi_{q_1}(x_i), \ldots, \chi_{q_m}(x_i)), (\chi_{q_1}(x_j), \ldots, \chi_{q_m}(x_j))\rangle = 0$, i.e, the i^{th} and j^{th} rows of $H_{S,Q}$ are orthogonal. Since i and j where arbitrarily chosen, $H_{S,Q}$ is a Hadamard matrix. □

Corollary 10. *When Q is a subspace-like set derived from a subspace S, a uniform training set for C_0^Q is S. When Q is a subspace, then any set Q' that is subspace-like, derived from Q is a uniform training set for C_0^Q. Notice that all these training sets have size exactly $|Q|$. Finally, when Π is the set $\{Q : Q$ is a subspace$\}$, the class C_0^{Π} is AAC-learnable exactly with subclass queries, in time $O(|Q|^4)$.*

Proof. If $Q = \{q_1, \ldots q_m\}$ is a subspace-like set derived from a subspace $S = \{x_1, \ldots, x_m\}$, it follows from Theorem 9 that $H_{Q,S} = [h_{ij}]$, where $h_{ij} = \chi_{x_j}(q_i)$ is Hadamard. Therefore $H_{S,Q} = H_{Q,S}^T$ is Hadamard and consequently invertible. Thus the system

$$H_{S,Q} \begin{bmatrix} \hat{f}(q_1) \\ \vdots \\ \hat{f}(q_m) \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}$$

has a unique solution. I.e. S is a training set for C_0^Q.
If $Q = \{q_1, \ldots, q_m\}$ is a subspace and $Q' = \{x_1, \ldots, x_m\}$ is subspace-like, derived from Q, then by Theorem 9 $H_{Q',Q}$ is Hadamard and consequently invertible. Thus

$$H_{Q',Q} \begin{bmatrix} \hat{f}(q_1) \\ \vdots \\ \hat{f}(q_m) \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}$$

has a unique solution. i.e. Q' is a training set for C_0^Q.
Finally note that by Proposition 7 Q' can be found in time $O(|Q|^4)$. Thus the class C_0^{Π} is AAC-learnable *exactly* with subclass queries, in time $O(|Q|^4)$. □

4 The Class $C_{\epsilon}^{Q,2}$

If f is a function selected from some class C and we hope to learn f, then it is required that we produce an algorithm A, i.e. a function $A : \mathbb{F}_2^n \to \{0, 1\}$ such that $f(x) = A(x)$ for most if not all $x \in \mathbb{F}_2^n$. The following definition identifies those algorithms A which are hypotheses for a function f with error bounded by some constant c.

Definition 11. Let $f : \mathbb{F}_2^n \to \{0,1\}$ and $A : \mathbb{F}_2^n \to \{0,1\}$. If $\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} |f(x) - A(x)| \leq c$, where $0 \leq c \leq 1$, then A is said to be a hypothesis for f with error bounded by c.

We show that asymptotically $C_\epsilon^{Q,2}$ has no reasonably small uniform training set if $\epsilon > 0$ and $|Q| \geq 2$, with $0^n \in Q$. I.e. if $C_{\epsilon,n}^{Q,2} = \{f : \mathbb{F}_2^n \to \{0,1\} | \exists g : \mathbb{F}_2^n \to \mathbb{R}$ such that spt $\hat{g} \subseteq Q$ and $\|f - g\|_2^2 \leq \epsilon\}$ and $S_n \subseteq \mathbb{F}_2^n$, then there exists $N \in \mathbb{N}$ such that if $n \geq N$, S_n can not be a uniform training set for $C_{\epsilon,n}^{Q,2}$.

Observation 12. *Let ϵ be positive and Q be a subset of $\mathbb{F}_2^n$ containing 0^n and of size at least 2. Let p be an arbitrary polynomial. There exists $N \in \mathbb{N}$ such that if $n \geq N$, then $C_{\epsilon,n}^{Q,2}$ has no uniform training set of size $\leq p(n)$ which yields a hypothesis error bounded by a fixed constant $c < 1/4$.*

Proof. Since $p(n)/2^n \to 0$ as $n \to \infty$, there exists $N \in \mathbb{N}$ such that if $n \geq N$, then $p(n)/2^n < \min\{\epsilon, 1/4 - c\}$. Let $n \geq N$ and suppose S is a uniform training set for $C_{\epsilon,n}^{Q,2}$, $|S| \leq p(n)$, and S yields a hypothesis error bounded by $c < 1/4$.

Since $|Q| \geq 2$, Q has a nonzero element, q. Note that χ_{0^n} and $c_q = \frac{1}{2}\chi_{0^n} + \frac{1}{2}\chi_q$ belong to $C_0^{Q,2} \subseteq C_{\epsilon,n}^{Q,2}$. Let $f : \mathbb{F}_2^n \to \{0,1\}$ be defined by $f(x) = \begin{cases} 1 & \text{if } x \in S \\ c_q & \text{if } x \in \mathbb{F}_2^n \setminus S \end{cases}$. Then $\|f - c_q\|_2^2 \leq |S|/2^n \leq p(n)/2^n < \epsilon$. Thus $f \in C_{\epsilon,n}^{Q,2}$.

Let A be the hypothesis for f obtained by sampling on S. Notice that since $f = \chi_{0^n}$ on S, A is also the hypothesis for χ_{0^n} obtained from S. Since $|\{x \in \mathbb{F}_2^n \setminus S | f(x) = c_q(x) \neq \chi_{0^n}(x)\}| \geq 2^{n-1} - |S| \geq 2^{n-1} - p(n)$, it follows that either $|\{x \in \mathbb{F}_2^n | f(x) \neq A(x)\}| \geq 2^{n-2} - p(n)$ or $|\{x \in \mathbb{F}_2^n | \chi_{0^n}(x) \neq A(x)\}| \geq 2^{n-2} - p(n)$. Thus either $\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} |f(x) - A(x)| \geq \frac{1}{4} - \frac{p(n)}{2^n} > c$ or $\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} |\chi_{0^n}(x) - A(x)| \geq \frac{1}{4} - \frac{p(n)}{2^n} > c$. Therefore the training set S yields a hypothesis A which produces an error greater than c in the case of either f or χ_{0^n}. Contradiction. □

5 The Class $C_\epsilon^{Q,\infty}$

We prove a theorem in the case where Q is subspace-like using duality, which gives us small, uniform training sets such that by processing the values of $f \in C_\epsilon^{Q,\infty}$ on these sets appropriately, the values $\hat{f}(x) : x \in Q$ can be estimated with small error. We then use the fact that $C_\epsilon^{Q,\infty} \subseteq C_{\epsilon^2}^{Q,2}$ and apply Theorem 4 to get the required AAC learning results.

Theorem 13. *Let Q be subspace-like. Given a function $f \in C_\epsilon^{Q,\infty}$, for each $q \in Q$, there is a function s_q such that $|\sum_x f(x)s_q(x) - \hat{f}(q)| \leq 2\epsilon$. Moreover, for each q, $|spt\ s_q|$ is exactly $|Q|$. We assume here that $|Q| \leq 2^{n-1}$, i.e, $|Q| \leq |\bar{Q}|$, which is true for most learning applications.*

Proof. We will show the existence of a function s_q such that $||s_q - 1/2^n\chi_q||_1 \leq 2$, $|\text{spt } s_q| = |Q|$; and $s_q - 1/2^n\chi_q \in C_0^{\bar{Q}}$. It will follow by the Duality Theorem 2

that $|\sum_x f(x)(s_q(x) - 1/2^n\chi_q(x))| = |\sum_x f(x)s_q(x) - \hat{f}(q)| \leq 2\epsilon$, thereby proving the theorem.

Let $Q = \{q_1, \ldots, q_{|Q|}\}$ be subspace-like. Note that $g \in C_0^{\bar{Q}}$ if and only if $\sum_x g(x)\chi_{q'}(x) = 0$ for all $q' \in Q$. Thus in order to find an s_q (for each $q \in Q$) such that $s_q - 1/2^n\chi_q \in C_0^{\bar{Q}}$, we can solve a system of $|Q|$ equations, one for each $q' \in Q$, of the form:

$$\sum_x (s_q(x) - 1/2^n\chi_q(x))\chi_{q'}(x) = 0$$

or

$$\sum_x s_q(x)\chi_{q'}(x) = \sum_x 1/2^n\chi_{q+q'}(x) = \begin{cases} 0 & \text{if } q' \neq q \\ 1 & \text{if } q' = q \end{cases}$$

for the 2^n variables, $s_q(x)$, $x \in \mathbb{F}_2^n$.

Let $S = \{x_1, \ldots, x_{|Q|}\}$ be a subspace of $\mathbb{F}_2^n$ which derives the subspace-like set Q. Then by Theorem 9, $H_{Q,S}$ is Hadamard. Furthermore, if $\mathbb{F}_2^n = \{x_1, \ldots, x_{|Q|}, \ldots, x_{2^n}\}$. then for each $q_i \in Q$ the system described above can be represented as:

$$[H_{Q,S} \;\; H_{Q,\bar{S}}] \begin{bmatrix} s_{q_i}(x_1) \\ \vdots \\ s_{q_i}(x_{2^n}) \end{bmatrix} = K_{q_i}$$

where K_{q_i} is a $|Q| \times 1$ matrix with i^{th} entry 1 and all other entries 0. Setting $s_{q_i}(x_{m+1}) = s_{q_i}(x_{m+2}) = \ldots = s_{q_i}(x_{2^n}) = 0$, we get:

$$\begin{bmatrix} s_{q_i}(x_1) \\ \vdots \\ s_{q_i}(x_{|Q|}) \end{bmatrix} = H_{Q,S}^{-1} \cdot K_{q_i}.$$

Since each entry in $H_{Q,S}^{-1}$ has absolute value $1/|Q|$, $|s_{q_i}(x_j)| = 1/|Q|$, for $1 \leq j \leq |Q|$. Thus for each $q \in Q$, $\|s_q - 1/2^n\chi_q\|_1 \leq 2$ and $|\text{spt } s_q| = m = |Q|$. Thereby proving the theorem. □

Corollary 14. *If Q is subspace-like, then the class $C_\epsilon^{Q,\infty}$ is AAC learnable in time $O(|Q|^2)$ with hypothesis error bounded by $O(|Q|\epsilon^2)$. Moreover, for the set $\Pi = \{Q : Q \text{ is a subspace}\}$, $C_\epsilon^{\Pi,\infty}$ is AAC learnable with subclass queries in time $O(|Q|^4)$, and hypothesis error bounded by $O(|Q|\epsilon^2)$.*
When Q is subspace-like and derived from a subspace S, then each spt s_q can be chosen as the subspace S, and when Q is itself a subspace, then any set Q' that is subspace-like and derived from Q can be used as spt s_q. All of the resulting training sets for $C_\epsilon^{Q,\infty}$, with Q subspace-like, have size $|Q|$.

Proof. For Q subspace-like, derived from the subspace S, the proof of Theorem 13 yields sampling distributions s_q and the uniform training set S, such that by sampling the function $f \in C_\epsilon^{Q,\infty}$ on this training set, one can estimate $\hat{f}(q)$ for $q \in Q$ to within 2ϵ. Thus if the functions s_q are known for each $q \in Q$, then

each coefficient $\hat{f}(q)$ can be calculated in $O(|Q|)$ time, providing a hypothesis, $\sum_{q\in Q}\hat{f}(q)\chi_q$, which can be calculated in time $O(|Q|^2)$.

Let Q, subspace-like, be fixed and note that $C_\epsilon^{Q,\infty}\subseteq C_{\epsilon^2}^{Q,2}$. Thus if $f\in C_\epsilon^{Q,\infty}$, then by Theorem 4, $\|\sum_{x\in Q}\hat{f}(x)\chi_x - f\|_2^2\leq\epsilon^2$. Therefore by Parseval's identity, $\sum_{x\in\bar{Q}}\hat{f}(x)^2\leq\epsilon^2$. So if g is the estimate to f obtained from the sampling distributions s_q, it follows that $\|f-g\|_2^2=\sum_{x\in\mathbb{F}_2^n}(\hat{f}(x)-\hat{g}(x))^2=\sum_{x\in Q}(\hat{f}(x)-\hat{g}(x))^2+\sum_{x\in\bar{Q}}\hat{f}(x)^2\leq|Q|\cdot(2\epsilon)^2+\epsilon^2=O(|Q|\epsilon^2)$.

If $Q\in\Pi$, then by Proposition 7 a subspace-like set S derived from Q can be computed in $O(|Q|^4)$ time. By Theorem 9 $H_{S,Q}$ is Hadamard and consequently so too is $H_{Q,S}$. Thus by Theorem 13, S can be used as a training set (i.e. spt $s_q=S,\forall q\in Q$) to learn the member of $C_\epsilon^{\Pi,\infty}$ in question, with hypothesis error bounded by $O(|Q|\epsilon^2)$.

The last part of the corollary follows directly from Theorem 9 and the proof of Theorem 13. □

Remark. Notice that s_q for $q=0^n$ has the property that $\sum_x f(x)s_q(x)$ is close (within 2ϵ) to the expected value of f, i.e $\hat{f}(0^n)=1/2^n\sum_x f(x)$. Hence, s_{0^n} (and also s_q for other q's by Corollary 14), could be chosen as any pseudorandom distribution that looks uniform to f and fools f. In other words, any set of pseudorandom strings for $C_\epsilon^{Q,\infty}$, where Q is subspace-like, can be used as spt s_q.

Notice that $C_\epsilon^{Q,\infty}\subseteq C_{\epsilon^2}^{Q,2}$, but they behave quite differently in that the former has uniform training sets as shown above, but the latter does not, as observed in 12.

6 Open Problems and Conjectures

(1) Extend the result of Theorem 13 to include general subsets, Q of $\mathbb{F}_2^n$. At present the result holds only for subspace-like sets Q.

(2) Extend the learnability results in Corollary 10 and and Corollary 14 when subclass queries are allowed. Currently the results only work for the class $\Pi=\{Q:Q\text{ is a subspace}\}$. The conjecture is that these results should work for a larger class, namely $\Lambda=\{Q:Q\text{ is subspace-like}\}$, for which a good structural description of of training sets is known from Theorem 9.

(3) As mentioned in the Introduction, the results here only consider weak learning in the AAC model with respect to the uniform distribution. Extend these results to the strong learning and distribution-independent models, perhaps with the use of boosting. The conjecture is that the learnability results in Corollary 14 should be extendible to give strong learning in the AAC model.

(4) All learnability results in the paper now involve to sets Q of *Parity functions* that are fixed or variable but obtainable by the learner using subclass queries. Extend these results to the case where Q is variable and unknown to the learner.

(5) Investigate the use of Corollary 14 in finding algorithms in the AAC model for functions with specific types of probabilistic communication protocols (as described in the Introduction). This would require investigating the properties of the Fourier spectra of the characteristic functions of combinatorial rectangles. The recent work of [5] may be applicable in this context.

(6) Put the model and results of this paper into the framework of learning in the presence of noise.

(7) Investigate the relationship of uniform training sets to VC dimension and ϵ-nets.

(8) When does AAC learnability (with a uniform training set) also imply PAC learnability and viceversa?

References

1. Alon, N., Vu, V. H.: Threshold gates, coin weighing, and indecomposable hypergraphs. FOCS (1996)
2. Angluin, D.: Learning regular sets from queries and counterexamples. Information and Computation (Vol. 75(2), 1987) 87–106
3. Angluin, D., Frazier, M., Pitt, L.: Learning conjunctions of Horn clauses. Machine Learning (Vol. 9, 1992) 147–164
4. Angluin, D., Hellerstein, L., Karpinski, M.: Learning read-once formulas with queries Journal of the ACM (Vol. 40, 1993) 185–210
5. Auer, P., Long, P., Srinivasan, A.: Pseudorandomness and learning of combinatorial rectangles. to appear in STOC (1997)
6. Ben-Or, M., Tiwari, P.: A deterministic algorithm for sparse multivariate polynomial interpolation. Proc. 20^{th} Ann. ACM Symp. Theory of Comput. (May 1988) 301–309
7. Bshouty, N. H.: Exact learning via the monotone theory. Proceedings of the 34th IEEE Symposium on the Foundations of Computer Science 302–311
8. Bshouty, N. H., Tamon, C.: On the Fourier spectrum of monotone functions. Proc. 27^{th} Ann. ACM Symp. Theory of Comput. (May 1995) 219–399
9. Bshouty, N., Mansour, Y.: Simple learning algorithms for decision trees and multivariate polynomials. Proc. 36^{th} Ann. IEEE Symp. Foundations of C.S. (1995) 304–311
10. Buck, R. C.: Applications of duality in approximation theory. Approximation of functions, Elsevier, H.L. Garabedian ed. (1964) 27–42
11. Enflo, P., Sitharam, M.: Stability of basis families and complexity lower bounds. ECCC report (Sept. 1996) Preprint also available at: http //nimitz.mcs.kent.edu/~ sitharam
12. Furst, M., Jackson, J., Smith, S.: Improved learning of AC^0 functions. 4^{th} Conf. on Computational Learning Theory, (1991) 317–325

13. Goldman, M., Hastad, J., Razborov, A. A.: Majority gates vs. general weighted threshold gates. 32^{nd} Ann. IEEE Symp. Foundations of C.S. (1991)
14. Gotsman, C., Linial, N.: Equivalence of two problems on the cube - A note. J. Comb. Theory, Ser. A **61** (1992) 142–146
15. Grolmusz, V.: Harmonic analysis, real approximation and communication complexity of Boolean functions. Manuscript (1994)
16. Hastad, J.: Computational limitations of small depth circuits. Ph. D thesis, MIT press (1986)
17. Hastad, J.: On the size of weights for threshold gates. SIAM J. Disc. Math. (1994) 484–492
18. Hajnal, A., Maass, W., Pudlák, P., Szegedy, M., Turán, G.: Threshold circuits of bounded depth. 28^{th} Ann. IEEE Symp. Foundations of C.S. (1987) 99–110
19. Jackson, J.: An efficient membership query algorithm for learning DNF with respect to the uniform distribution. Proc. 35^{th} Ann. IEEE Symp. Foundations of C.S. (1994) 42–53
20. Kushilevitz, E., Mansour, Y.: Learning decision trees using the Fourier transform. 32^{nd} Ann. IEEE Symp. Foundations of C.S. (1991) 455–464
21. Linial, N., Mansour, Y., Nisan, N.: Constant depth circuits, Fourier transforms, and learnability. 30^{th} Ann. IEEE Symp. Foundations of C.S. (1989) 574–579
22. Nisan, N., Szegedy, M.: On the degree of Boolean functions as real polynomials. 24^{th} Ann. ACM Symp. on Theory of Computing (1992) 462–467
23. Paturi, R.: On the degree of polynomials that approximate symmetric Boolean functions. 24^{th} Ann. ACM Symp. on Theory of Computing (1992) 468–474
24. Sitharam, M.: Pseudorandom generators and learning algorithms for AC^0. Ann. ACM Symp. on Theory of Computing (1994) 478–488
25. Schapire, R., Sellie, L.: Learning sparse multivariate polynomials over a field with queries and counterexamples. Proceedings of the 6th Workshop on Computational Learning Theory 17–26
26. Sitharam, M.: Approximation from spaces of functions over the cube, complexity lower bounds, derandomization, and learning algorithms. See ECCC reports. Preprint also available at: http //nimitz.mcs.kent.edu/~ sitharam