On the Fourier Spectrum of Symmetric Boolean Functions with Applications to Learning Symmetric Juntas

Richard J. Lipton*
Georgia Institute of Technology
College of Computing
Atlanta, GA 30332, USA
rjl@cc.gatech.edu

Evangelos Markakis
Georgia Institute of Technology
College of Computing
Atlanta, GA 30332, USA
vangelis@cc.gatech.edu

Aranyak Mehta
Georgia Institute of Technology
College of Computing
Atlanta, GA 30332, USA
aranyak@cc.gatech.edu

Nisheeth K. Vishnoi
IBM India Research Lab
Block 1, IIT Delhi
New Delhi, 110016, India
nvishnoi@in.ibm.com

Abstract

We study the following question:

> *What is the smallest t such that every symmetric boolean function on k variables (which is not a constant or a parity function), has a non-zero Fourier coefficient of order at least 1 and at most t?*

We exclude the constant functions for which there is no such t and the parity functions for which t has to be k. Let $\tau(k)$ be the smallest such t. The main contribution of this paper is a proof of the following self similar nature of this question:

$$\text{If } \tau(l) \leq s, \text{ then for any } \epsilon > 0 \text{ and for } k \geq k_0(l, \epsilon), \quad \tau(k) \leq \left(\frac{s+1}{l+1} + \epsilon\right) k$$

Coupling this result with a computer based search which establishes $\tau(30) = 2$, one obtains that for large enough k, $\tau(k) \leq 3k/31$.

The motivation for our work is to understand the complexity of learning symmetric juntas. A k-junta is a boolean function of n variables that depends only on an unknown subset of k variables. If f is symmetric in the variables it depends on, it is called a symmetric k-junta. Our results imply an algorithm to learn the class of symmetric k-juntas, in the uniform PAC learning model, in time approximately $n^{\frac{3k}{31}}$. This improves on a result of Mossel, O'Donnell and Servedio in [11], who show that symmetric k-juntas can be learned in time $n^{\frac{2k}{3}}$ (the main result in [11] is much more general, giving a bound of $n^{0.7k}$ for learning juntas).

Technically, the study of $\tau(k)$ is equivalent to the study of $0/1$ solutions of a system of Diophantine equations involving binomial coefficients. As a first step, we simplify these Diophantine equations by moving to a representation of boolean functions, which is equivalent to their Fourier representation, but seems much simpler for the application of number theoretic tools. Once this is done, we reduce these equations modulo carefully chosen prime numbers to get a simpler system of equations which we can analyze. Finally, we combine the information about the equations over the finite fields in a combinatorial manner to deduce the nature of the $0/1$ solutions.

*Research supported by NSF grants CCR-0002299 and CCF-0431023.

1 Introduction

Problem statement

The study of the Fourier representation of boolean functions has proved to be extremely useful in computational complexity and learning theory. In this paper we focus on the Fourier spectrum of symmetric boolean functions and we study the following question:

> What is the smallest t such that every symmetric boolean function on k variables (which is not a constant or a parity function), has a non-zero Fourier coefficient of order at least 1 and at most t?

We exclude the two constant functions, for which there is no such t, and the two parity functions, for which t has to be k. Let $\tau(k)$ be the smallest such t. While the above question is interesting in its own right, there is also an important learning theory application behind it, which we outline next.

Motivation

The motivation to study $\tau(k)$ comes from the following fundamental problem in computational learning theory: learning in the presence of irrelevant information. In many practical applications (e.g. feature recognition) an observed function may be a result of very few factors which are hidden in a large array of irrelevant information. One formalization of the problem is as follows: We want to learn an unknown boolean function of n variables, which depends only on $k \ll n$ variables. Typically, k is $O(\log n)$. Such a function is referred to as a k-junta. We are provided with a set of labeled examples $\langle \mathbf{x}, f(\mathbf{x}) \rangle$, where the $\mathbf{x}$'s are picked uniformly and independently at random from the domain $\{0,1\}^n$. We wish to identify the k relevant variables and the truth table of the function.

The problem was first posed by Blum [1] and Blum and Langley [4], and it is considered [2, 11] to be one of the most important open problems in the theory of uniform distribution learning. It has connections with learning DNF formulas and decision trees of super-constant size, see [5, 8, 10, 13, 14] for more details. The general case is believed to be hard and has even been used in the construction of a cryptosystem [3]. A trivial algorithm runs in time roughly n^k by doing an exhaustive search over all possible sets of relevant variables. Two important classes of juntas are learnable in polynomial time: parity and monotone functions. Learning parity functions can be reduced to solving a system of linear equations over $\mathbb{F}_2$ [7]. Monotone functions have non-zero singleton Fourier coefficients (see [11]). For the general case, the first significant breakthrough was given in [11] - learning with confidence $1-\delta$ in time $n^{0.7k}\mathrm{poly}(\log 1/\delta, 2^{\mathrm{k}}, \mathrm{n})$. Note that we allow the running time to be polynomial in 2^k, since this is the size of the truth-table which is output. In the typical setting of $k = O(\log n)$, this becomes polynomial in n.

Fourier based techniques in learning were introduced in [9] and have proved to be very successful in several problems. One reason for this success is that Fourier coefficients are easy to compute in the uniform distribution learning model. Furthermore, if a Fourier coefficient is non-zero then its entire support is contained in the set of relevant variables. Hence, it is interesting to ask: what are the subclasses of juntas for which Fourier based techniques yield fast learning algorithms? An important and natural subclass is the class of symmetric juntas. While this subclass contains only 2^{k+1} functions, the problem is not known to be significantly easier than the general case. The bound before our work was $n^{2k/3}$ [11], which is not much better than the best bound for general juntas (also obtained in [11]). Our results imply an improved bound for learning symmetric juntas via the Fourier based algorithm.

We believe that the case of symmetric juntas constitutes a good "challenge problem" towards the goal of learning general juntas. One motivation for this is a consideration of the following well-known challenge problem [2] :
Let $f(x_1,\ldots,x_n) := \mathrm{MAJORITY}(x_1,\ldots,x_{2k/3}) \bigoplus \left(x_{2k/3+1} \oplus \cdots \oplus x_k\right)$, where $x_1,\ldots,x_k$ are some *unknown* variables among $x_1,\ldots,x_n$. This subclass has been identified as a candidate hard to learn class [2]. The current bound for learning this subclass of juntas is $n^{k/3}$, and it is asked in [2] if a faster algorithm exists. Note that f is invariant under permutations of $\{x_1,\ldots,x_{2k/3}\}$ and under permutations of $\{x_{2k/3},\ldots,x_k\}$, i.e., it is invariant under a large group of symmetries. This suggests that it is interesting to begin with the case of symmetric juntas.

Our results

Our main result is:

$$\text{If } \tau(l) \le s, \text{ then for any } \epsilon > 0 \text{ and for } k \ge k_0(l,\epsilon),$$
$$\tau(k) \le \left(\tfrac{s+1}{l+1} + \epsilon\right) k$$

Coupling this result with a computer based search which establishes $\tau(30) = 2$, one obtains that for large enough k, $\tau(k) \le 3k/31$.

Our result implies a bound of $n^{3k/31}$ for the Fourier based learning algorithm for the class of symmetric k-juntas. To our knowledge, this is the best known upper bound for learning symmetric juntas under the uniform distribution. Independent of the learning problem, the fact that symmetric boolean functions have non-zero Fourier coefficients of relatively small order provides new insight into the structure of these functions.

Techniques

The study of $\tau(k)$ is equivalent to the study of $0/1$ solutions of a system of Diophantine equations involving binomial coefficients. As a first step, we simplify these Diophantine equations by moving to a representation which is equivalent to the Fourier representation, but seems much simpler for the application of number theoretic tools. Once this is done, we reduce the equations modulo carefully chosen prime numbers to get a simpler system of equations which we can analyze. Finally, we combine the information about the equations over the finite fields in a combinatorial manner to deduce the nature of the $0/1$ solutions.

The specific bound of $3k/31$ is then obtained by a computer search. The following well-known self-similarity property of Pascal's Triangle plays an important role: If $m = lp$ for some integer l and some prime p, then the nonzero values obtained by reducing the binomial coefficients of the m-th row of Pascal's Triangle modulo p, can be read off directly from the l-th row of Pascal's Triangle.

A remark concerning the computer search: An interesting aspect of the proof is the fact that if we can find an explicit value l, for which all symmetric l-juntas have non-zero Fourier coefficients of very small order, then we can prove that for all large k, k-juntas have non-zero Fourier coefficients of relatively small order. To find an explicit value for l, we did a computer search on all symmetric boolean functions of up to 30 bits and computed their Fourier coefficients of small order. The search revealed that every symmetric function on 30 bits (which is not parity or constant) has a nonzero Fourier coefficient of order 1 or 2. In our proof we use this fact to obtain the bound of $3k/31$ for large enough k. A search beyond 30 bits may yield better bounds.

Related work

Previously, the idea of reducing binomial coefficients modulo a prime number has been used in [15] to prove lower bounds on the degree of polynomials representing symmetric boolean functions. In [15], their problem reduces to showing that a certain sum of binomial coefficients is non-zero, which is done by reducing the sum modulo a prime number. Our problem involves a collection of sums which we have to prove are unequal. For this we need to consider reductions modulo two different carefully chosen primes and combine the information obtained by the two reductions in a combinatorial manner.

The result of [15] has in fact been used in the proof of the previous best $n^{2k/3}$ bound for learning symmetric juntas [11]. Using [15], it is shown in [11] that if a symmetric function f is *balanced*, i.e., $Pr[f(x) = 1] = 1/2$, then it has a non-zero Fourier coefficient of order $o(k)$. The $2k/3$ bottleneck comes in the case of *unbalanced* symmetric functions, which are analyzed through a different argument. As noted in [11], the result of [15] does not seem to be applicable to learning unbalanced functions.

2 Fourier Coefficients of Boolean Functions

2.1 Notation

We consider boolean functions from $\{0,1\}^k \to \{0,1\}$. For a set $S \subseteq [k]$, define $\chi_S : \{0,1\}^k \to \{1,-1\}$ to be the function $\chi_S(\mathbf{x}) := (-1)^{\sum_{i \in S} x_i}$ (by convention, the boldface $\mathbf{x}$ denotes the vector $(x_1, \ldots, x_k)$). For a function $f : \{0,1\}^k \to \{0,1\}$, and $S \subseteq [k]$, define the *Fourier coefficient* corresponding to S as $\hat{f}(S) := \frac{1}{2^k} \sum_{\mathbf{x} \in \{0,1\}^k} f(\mathbf{x}) \chi_S(\mathbf{x})$. The *order* of a Fourier coefficient $\hat{f}(S)$ is $|S|$. The Fourier expansion of f is: $f(\mathbf{x}) = \sum_{S \subseteq [k]} \hat{f}(S) \chi_S(\mathbf{x})$.

If f is symmetric, f is completely determined by its value on any $k+1$ vectors of distinct weights, where the *weight* of a boolean vector is the number of 1's in it. We will use the following vector representation of f: $\nu(f) := (f_0, f_1, \ldots, f_k)^T$. Here f_i is the value of f on a vector of weight i. Further f has precisely $k+1$ (non-equivalent) Fourier coefficients, $(\hat{f}_0, \ldots, \hat{f}_k)$. Here $\hat{f}_t$ is defined as $\hat{f}(S)$, for some $S \subseteq [k]$ with cardinality t. Since f is symmetric, this does not depend on the choice of S. The following four special symmetric functions on k variables will appear often: the two constant functions $\mathbf{0}$ and $\mathbf{1}$, the parity function $\oplus$, and its complement $\overline{\oplus}$.

2.2 An equivalent formulation as a Diophantine problem

In this section we give an equivalent condition for the existence of a non-zero Fourier coefficient of a boolean function f. While we prove the equivalence for all boolean functions, we use it only for the special case of symmetric functions.

Let $f : \{0,1\}^k \mapsto \{0,1\}$ be a boolean function. For a vector $\mathbf{x} = (x_1, \ldots, x_k)$, and a set $S \subseteq [k]$, let $\mathbf{x}_S$ be the projection of $\mathbf{x}$ on the indices of S. Let $\sigma \in \{0,1\}^{|S|}$. Define the following probabilities:

$$p_{S,\sigma}(f) := \Pr[f(\mathbf{x}) = 1 \mid \mathbf{x}_S = \sigma]$$

Unless mentioned, all probabilities are over the uniform distribution. For $t \geq 1$, call a boolean function f on k variables *t-null*, if for all sets $S \subseteq [k]$, with $|S| = t$, and for all $\sigma \in \{0,1\}^t$, the probabilities $p_{S,\sigma}(f)$ are all equal to each other. The following lemma reveals the connection with the Fourier coefficients of f.

Lemma 1 *Let f be a boolean function on k variables. f is t-null for some $1 \leq t \leq k$, if and only if, for all $\emptyset \neq S \subseteq [k]$ with cardinality at most t, $\hat{f}(S) = 0$.*

Proof : It can be easily verified that if f is t-null, then for all $\emptyset \neq S \subseteq [k]$ with cardinality at most t, $\hat{f}(S) = 0$. This follows from the fact that the Fourier coefficients of order at most t can be expressed as ± 1 combinations of $p_{S,\sigma}(f)$ with $\sigma \in \{0,1\}^t$, and $S \subseteq [k], |S| = t$. When f is t-null, the terms cancel out. The proof of the other direction is by induction and we omit it here. □

The following is an immediate corollary of this lemma.

Corollary 2 *Let f be a boolean function on k variables. If f is t-null for some $1 \leq t \leq n$ then f is s-null for $1 \leq s \leq t$.*

When we consider the case of symmetric functions, $p_{S,\sigma}(f)$ just depends on $s := |S|$ and the weight w of σ. We denote this by $p_{s,w}(f)$. It is clear that:

$$p_{s,w}(f) = \frac{1}{2^{k-s}} \sum_{i=0}^{k} f_i \binom{k-s}{i-w}$$

where $\binom{l}{m}$ is 0 if $m < 0$ or $m > l$, and $\binom{0}{0}$ is 1. By definition, f is s-null if for $0 \leq w \leq s$, $p_{s,w}(f)$ are all equal. Hence, f is s-null iff there exists $c := c(f,s,k)$ such that

$$\sum_{i=0}^{k} \binom{k-s}{i-w} f_i = c, \ \ \forall\, 0 \leq w \leq s. \tag{1}$$

Thus we have:

Lemma 3 *For $1 \leq s \leq k$, let $A_{k,s}$ be the $(s+1) \times (k+1)$ matrix:*

$$A_{k,s}(j,i) := \binom{k-s}{i-j}$$

A symmetric function f is s-null if and only if there exists a positive integer $c := c(f,s,k)$ such that:

$$A_{k,s} \cdot \nu(f) = c\mathbf{1}$$

It is easy to see that the constant boolean functions $\{\mathbf{0}, \mathbf{1}\}$ satisfy this system of equations for all s, i.e. they are s-null for all s, s.t. $1 \leq s \leq k$. One can also see that the boolean functions $\{\oplus, \overline{\oplus}\}$ are s-null for all s s.t. $1 \leq s < k$. From Lemma 1 and Lemma 3 we get:

Corollary 4 *All symmetric boolean functions $f \notin \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$ have a non-zero Fourier coefficient of order at most s_0 (and at least 1) iff there exists s, $1 \leq s \leq s_0$ s.t. $\{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$ are the only solutions to*

$$\sum_{i=0}^{k-s} f_i \binom{k-s}{i} = \ldots = \sum_{i=s}^{k} f_i \binom{k-s}{i-s} \tag{2}$$

The question is how large must s_0 in the statement of Corollary 4 be. In the next section, we show that $s_0 \leq \frac{3}{31}k$ for large enough k.

3 The Main Result

3.1 Number theoretic preliminaries

We will first present some facts that we are going to use in proving our main theorem. The next easy result is a special case of Lucas' Theorem [6] and illustrates the *self similar* nature of the Pascal's Triangle modulo primes.

Lemma 5 *For a prime p, an integer $l \geq 0$ and $0 \leq i \leq lp$, $\binom{lp}{i} \equiv \binom{l}{j} \mod p$ if $i = jp$ for some $0 \leq j \leq l$, and 0 otherwise.*

Proof : For a prime p, and $0 \leq j \leq p$, $\binom{p}{j} \equiv 1 \mod p$ if $j = 0$ or $j = p$, and $\binom{p}{j} \equiv 0 \mod p$ otherwise. Hence, for an indeterminate x, $(1+x)^p \equiv 1 + x^p \mod p$. Consider $\sum_{i=0}^{lp} \binom{lp}{i} x^i = (1+x)^{lp} = ((1+x)^p)^l$. Reducing this sum modulo p, and using the fact above, one obtains

$$\sum_{i=0}^{lp} \binom{lp}{i} x^i \equiv (1+x^p)^l \mod p.$$

But $(1+x^p)^l = \sum_{j=0}^{l} \binom{l}{j} x^{pj}$. Comparing coefficients of x^{pj} on both sides of the above equation, one gets the desired conclusion. □

On numerous occasions, we will use the following result about the density of primes. This follows from the Prime Number Theorem.

Lemma 6 *For large enough n, there is a prime $p \leq n$, such that $p = n - o(n)$.*

3.2 The case of $k/2$

In this section we give a self-contained proof of the following (weaker) result. The aim is to illustrate the key ideas behind the proof of Theorem 9.

Theorem 7 *For any symmetric boolean function f on k variables with $f \notin \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$, there exists $1 \leq t \leq \frac{k}{2} + o(k)$ such that $\hat{f}_t \neq 0$.*

We need the following combinatorial lemma. For positive integers k, p, q, s.t. $p \neq q$, let $G_{k,p,q}$ be the graph with vertex set $\{0, 1, 2, \ldots, k\}$, and the edge set $\{(i,j) : |i-j| = p \text{ or } q\}$.

Lemma 8 *For positive integers k, p, q such that $(p,q) = 1$ and $p + q \leq k$, $G_{k,p,q}$ is connected.*

Proof : We proceed by induction on $p + q$. Without loss of generality, let $p > q$. Clearly, the lemma holds for the base case. Let i, j be s.t. $0 \leq i < j \leq k$ and $j - i = p - q$. Since $p+q \leq k$, either $i+p \leq k$ or $i-q \geq 0$. In either case, there is a path of length 2 between i and j. Hence replacing the edges $\{(u,v) : |u-v| = p\}$ by the new edges $\{(u',v') : |u'-v'| = p-q\}$ does not increase the connectivity of the graph. It suffices to show that $G_{k,p-q,q}$ is connected, which follows by the induction hypothesis. □

Proof of Theorem 7 : Let f be a symmetric function such that for every $1 \leq t \leq \frac{k}{2} + o(k)$, $\hat{f}_t = 0$. We will show that $f \in \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$.

By Lemma 6, we can pick primes p, q, s.t. $\frac{k}{2} - o(k) = p < q \leq \frac{k}{2}$. Since $k-p$ and $k-q$ are both at most $\frac{k}{2}+o(k)$, we get from Lemma 1 that f is $(k-p)$-null and $(k-q)$-null.

Hence, by Lemma 3, $\exists\, c_1, c_2$ such that

$$A_{k,k-p}\nu(f) = c_1\mathbf{1} \quad \text{and} \quad A_{k,k-q}\nu(f) = c_2\mathbf{1}$$

Consider these two systems of equations modulo p and q respectively. Let $0 \leq c_p < p$ and $0 \leq c_q < q$ be s.t. $c_p \equiv c_1 \mod p$, and $c_q \equiv c_2 \mod q$. We will use $\equiv_p$ to denote congruences $\mod p$ (and similarly for q). The systems become:

$$A_{k,k-p}\nu(f) \equiv_p c_p\mathbf{1} \quad \text{and} \quad A_{k,k-q}\nu(f) \equiv_q c_q\mathbf{1}$$

Now, from Lemma 5, we see that $\binom{p}{i} \equiv_p 1$ if $i = 0$ or $i = p$, and $\binom{p}{i} \equiv_p 0$ otherwise (and similarly for q). Hence we see that the equations are of the form

$$f_i + f_{i+p} \equiv_p c_p \text{ for } 0 \leq i \leq k-p$$

and

$$f_i + f_{i+q} \equiv_q c_q \text{ for } 0 \leq i \leq k-q$$

Since $f_i \in \{0,1\}$ and $p > 2$, these modular equations are in fact exact equalities and $c_p, c_q \in \{0,1,2\}$. If $c_p = 0$ then it follows that $c_q = 0$ and $f = \mathbf{0}$ (because every variable f_i is present in at least one equation, since $p \leq k/2$). If $c_p = 2$ then $c_q = 2$ and $f = \mathbf{1}$. The only remaining case is $c_p = c_q = 1$. This gives

$$f_i = \bar{f}_{i+p} \text{ for } 0 \leq i \leq k-p$$

and

$$f_i = \bar{f}_{i+q} \text{ for } 0 \leq i \leq k-q$$

In other words, $|i-j| = p$ or q implies that $f_i = \bar{f}_j$. Since $G_{k,p,q}$ is connected (Lemma 8) it follows that fixing the value of any f_i uniquely determines f, and hence, there are at most 2 possible choices for f. We can see that $\{\oplus, \overline{\oplus}\}$ are solutions to the equations, hence they are the only solutions in this case.

□

3.3 The main theorem

In this section we prove our main theorem. Recall that $\tau(k)$ is the smallest number t such that every symmetric boolean function f on k variables, with $f \notin \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$, has a non-zero Fourier coefficient of order at least 1 and at most t.

Theorem 9 *Let $0 < s \leq l$ be fixed integers such that $\tau(l) \leq s$ and let $\epsilon > 0$. There exists a constant $k_0 := k_0(l, \epsilon)$ such that, for all $k \geq k_0$, $\tau(k) \leq \left(\frac{s+1}{l+1} + \epsilon\right) k$.*

Proof :

Let f be a symmetric boolean function on k variables. Suppose that f is t-null, for all $t \leq \left(\frac{s+1}{l+1} + \epsilon\right) k$. We will show that $f \in \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$.

Let $m = l - s$. Assume that there is a prime p such that $k = (m+s+1)p - 1$. We handle the case when there is no such prime p later. Set $t := k - mp = (s+1)p - 1$. Since $p = \frac{k+1}{l+1}$,

$$t = \left(\frac{s+1}{l+1}\right) k + \frac{s+1}{l+1} - 1 \leq \left(\frac{s+1}{l+1}\right) k.$$

Hence, f is t-null and there is an integer c such that

$$A_{k,t}\nu(f) = c\mathbf{1}. \tag{3}$$

We remark that the role of ϵ is redundant in this case. It will play a role when we cannot choose p such that $k - t = mp$.

Reducing to a smaller problem

Note that, by definition of t, $k - t = mp$. For $0 \leq i \leq p-1$, let $\mathbf{F}_i := (f_i, f_{i+p}, f_{i+2p}, \ldots, f_{i+lp})$. Hence, reducing Equations (3) modulo p, and using Lemma 5, one obtains the following systems of equations.

$$\begin{aligned} A_{l,s}\mathbf{F}_0 &\equiv_p c'\mathbf{1} \\ A_{l,s}\mathbf{F}_1 &\equiv_p c'\mathbf{1} \\ &\vdots \\ A_{l,s}\mathbf{F}_{p-1} &\equiv_p c'\mathbf{1} \end{aligned}$$

Here $c' \equiv c \mod p$. By choosing k_0 large enough, we can ensure that for $k \geq k_0$, $p > 2^l$. In that case, the modular equations are in fact exact. That is, there is an integer $d \geq 0$, such that the following set of equations hold:

$$\begin{aligned} A_{l,s}\mathbf{F}_0 &= d\mathbf{1} \\ A_{l,s}\mathbf{F}_1 &= d\mathbf{1} \\ &\vdots \\ A_{l,s}\mathbf{F}_{p-1} &= d\mathbf{1} \end{aligned} \tag{4}$$

Using the fact that $\tau(l) \leq s$, we deduce that for any i, the system of equations $A_{l,s}\mathbf{F}_i = d\mathbf{1}$ has at most 4 solutions, namely the constant and parity solutions (when treating $\mathbf{F}_i$ as a symmetric function on l bits). This implies that there are at most 4^p choices for f. Now we show how to narrow down these choices to 4.

Combining the smaller instances

Let $\frac{k}{2} < mp \leq q \leq (m+1)p$ be a prime. Since f is t-null, and $t = k - mp \geq k - q$, by Corollary 2, f is $(k-q)$-null. Consider the system of equations $A_{k,k-q}\nu(f) = c\mathbf{1}$

modulo the prime q. As in the proof of Theorem 7, we get, for some $e \geq 0$, exact equations of the following form:

$$\begin{aligned} f_0 + f_q &= e \\ f_1 + f_{q+1} &= e \\ &\vdots \\ f_{k-q} + f_q &= e. \end{aligned} \tag{5}$$

The idea is that these equations, along with Equations (4), are sufficient to restrict f to one of the four functions, $\{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$, as desired. First, we need a simple fact. For an integer $r \geq 0$, let $(r)_p := r \mod p$. Also, for $0 \leq i \leq p-1$, let $[iq]_p := \{(iq)_p, (iq)_p + p, \ldots, (iq)_p + (m+s)p\}$.

Fact 10 *For $0 \leq i < j \leq p-1$, $[iq]_p \cap [jq]_p = \emptyset$.*

Now, fix $f_0, f_p \in \mathbf{F}_0$. As noticed before, this fixes all the variables in $\mathbf{F}_0$. Using Equations (5), in particular, we get that f_q and f_{q+p} are fixed. Now Equations (4) imply that all the indices in $\mathbf{F}_{(q)_p}$ get fixed. Iterating the alternate use of these two systems of equations, along with Fact 10, one obtains that all the variables in $\mathbf{F}_i$, for every i, are fixed, once f_0 and f_p are fixed. Hence, f has at most four choices: $\{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$, one for every possible fixing of $\{f_0, f_p\}$.

Since we need $p > 2^l$ and since $k = (l+1)p - 1$, we can choose $k_0 := k_0(l)$ such that for all $k \geq k_0$, $\tau(k) \leq t = \left(\frac{s+1}{l+1}\right) k + \frac{s+1}{l+1} - 1 \leq \left(\frac{s+1}{l+1}\right) k$.

Handling the residual class of variables

Now we consider the case when there is no prime p such that $k = (m+s+1)p - 1$. In this case, we pick a prime p in the interval $\left[\frac{k}{m+s+1} - o(k), \frac{k}{m+s+1}\right]$. We are guaranteed the existence of such a prime by Lemma 6. Let $t = k - mp$. Hence, $(s+1)p + o(p) \geq t \geq (s+1)p$. Since we think of m as a constant, $p = \Omega(k)$. Hence, there is a small number ($o(k)$) of variables, say $\mathbf{R}$, which remain to be dealt with in the previous argument. In particular $\mathbf{R} = \{f_{(l+1)p}, f_{(l+1)p+1}, ..., f_k\}$ and $\{f_0, ..., f_k\} = \left(\cup_{i=0}^{p-1} \mathbf{F}_i\right) \cup \mathbf{R}$. By the argument in the previous case, fixing f_0 and f_p fixes all the variables in $\cup_{i=0}^{p-1} \mathbf{F}_i$. Further, since $|\mathbf{R}| = o(k)$, and $q > k/2$, every variable in $\mathbf{R}$ appears in one of the Equations (5) along with a variable in $\cup_{i=0}^{p-1} \mathbf{F}_i$, and hence gets fixed.

As before, we need to ensure that $p > 2^l$. Since $p = \frac{k}{l+1} - o(k)$, we can choose, for every $\epsilon > 0$, large enough $k_0 := k_0(l, \epsilon)$, such that for all $k \geq k_0$, $\tau(k) \leq t \leq \left(\frac{s+1}{l+1} + \epsilon\right) k$. This completes the proof of Theorem 9.

□

3.4 An explicit bound on $\tau(k)$

Using a computer search, we verified (among others) that $\tau(30) = 2$. Plugging this in Theorem 9, we obtain the following Corollary:

Corollary 11 *Let $\epsilon > 0$. There exists a constant $k_0 := k_0(\epsilon)$ such that, for all $k \geq k_0$, and for every symmetric boolean function f on k variables with $f \notin \{\mathbf{0}, \mathbf{1}, \oplus, \overline{\oplus}\}$, there is an integer $1 \leq t \leq \left(\frac{3}{31} + \epsilon\right) k$, such that f has a non-zero Fourier coefficient of order t.*

4 Learning symmetric juntas

In this section we apply Corollary 11 to obtain fast learning algorithms for the class of symmetric k-juntas on n variables. First we need some preliminaries and well known tools from computational learning theory.

4.1 Preliminaries

We consider the PAC learning model [12]. The learning problem at hand is a *Concept Class* $\mathcal{C} = \bigcup_n \mathcal{C}_n$, where each $\mathcal{C}_n$ is a collection of boolean functions from $\{0,1\}^n \to \{0,1\}$. Let ϵ be an *accuracy parameter* and δ a *confidence parameter*. A learning algorithm $\mathcal{A}$ for $\mathcal{C}$ has access to an *oracle* $\mathcal{I}(f)$ for $f \in \mathcal{C}_n$. A query to $\mathcal{I}(f)$ outputs a labeled example $\langle \mathbf{x}, f(\mathbf{x}) \rangle$, where $\mathbf{x}$ is drawn from $\{0,1\}^n$ according to some probability distribution. $\mathcal{A}$ is said to be a learning algorithm for the class $\mathcal{C}$ if for all $f \in \mathcal{C}$, when $\mathcal{A}$ is run with oracle $\mathcal{I}(f)$, it outputs, with probability at least $1 - \delta$, a hypothesis h such that $\Pr_{\mathbf{x}}[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \epsilon$. Although Valiant's PAC model is defined for general distributions, in this paper we will be concerned only with the uniform distribution.

We recall the definition of a k-junta. Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function. We say that f *depends* on the variable i, if there are vectors $\mathbf{x}$ and $\mathbf{y}$ that differ only on the i'th coordinate and $f(\mathbf{x}) \neq f(\mathbf{y})$. A function that depends only on an (unknown) subset of $k \ll n$ variables is called a k-junta. The variables on which f depends are called the *relevant* variables of f. Typically $k = O(\log n)$. Hence, a running time that is polynomial in $2^k, n$ and $\log(1/\delta)$ is considered efficient. A symmetric k-junta is a boolean function which is symmetric in the variables it depends on. The class of all such functions defined on n variables is the class of symmetric k-juntas. In this section, we present an algorithm for learning this class in the uniform PAC model.

4.2 Analysis of the Fourier based algorithm

We will use the following facts about learning in the PAC model which are well known.

(i) We can exactly calculate the Fourier coefficients of the target function with confidence $1-\delta$ in time $poly(\log 1/\delta,\ 2^k, n)$ using standard Chernoff-Hoeffding bounds (see [9, 11]).

(ii) We can decide whether the target function f is constant or not in time $poly(\log 1/\delta, 2^k)$.

(iii) We can learn a parity function in time $n^{\omega} poly(\log 1/\delta, 2^k)$ [7]. Here ω is the exponent for matrix multiplication, $\omega < 2.376$.

We state the standard Fourier based algorithm below:

Throughout the algorithm, we maintain a set of relevant variables, R.

- Check if the function is constant or parity.
- If not, set $R := \emptyset$, $t := 1$.
  1. For every subset of t variables, say $S = \{x_{i_1}, ..., x_{i_t}\}$ do:
     (a) Compute $\hat{f}(S)$.
     (b) If $\hat{f}(S) \neq 0$, then $R := R \cup S$.
  2. If for all sets S of size t, $\hat{f}(S) = 0$ then $t := t+1$ and go to step 1.
  3. Else, R now contains all the relevant variables. Draw enough samples to build f's truth table and halt.

If x_i is an irrelevant variable for f, then it is easy to see that for any S containing x_i, $\hat{f}(S) = 0$. Hence if $\hat{f}(S) \neq 0$, for some S, then S contains only relevant variables. Since the function is symmetric, for any two sets S, T of relevant variables such that $|S| = |T|$, we have $\hat{f}(S) = \hat{f}(T)$. Hence the first time that we will identify some relevant variables in the algorithm ($\hat{f}(S) \neq 0$ for some S, $|S| = s$), we will actually be able to identify all the relevant variables, and the running time will be roughly n^s. As a direct consequence of Corollary 11 we obtain the following Theorem:

Theorem 12 *The class of symmetric k-juntas can be learned exactly under the uniform distribution with confidence $1-\delta$ in time $n^{\frac{3k}{31}+o(k)} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$.*

5 Conclusion

The most important open problem that remains is to ascertain the true behavior of the function $\tau(k)$. It may even be that $\tau(k)$ is a constant for all k, but resolving this seems hard. A relatively easier problem, which seems approachable, is to show that $\tau(k)$ is at most a constant for infinitely many k. Using Theorem 9, this will already imply $\tau(k) \leq \epsilon k$, for all $\epsilon > 0$, for large enough k. Among other problems, it will be very useful to be able to determine $\tau(l)$ quickly for $l > 30$. Right now, we know of no method other than essentially an exponential algorithm.

6 Results of the Computer Search

The following table is based on a computational search for $\tau(l)$ for small values of l. The rows in the table correspond to values of l. The columns correspond to various values of s. The (l, s)-th entry of the table is the number of symmetric boolean functions f such that $A_{l,s}\nu(f)$ is a constant vector. Hence, whenever this entry is 4, $\tau(l) \leq s$. The least value of $\frac{s+1}{l+1}$ for which $\tau(l) \leq s$ is for $l = 30, s = 2$, giving the ratio $3/31$.

l	s=1	s=2	s=3	s=4
2	4	4	4	4
3	6	4	4	4
4	8	4	4	4
5	8	4	4	4
6	20	4	4	4
7	26	8	4	4
8	48	10	6	4
9	42	10	6	4
10	64	6	6	4
11	66	4	4	4
12	144	4	4	4
13	178	8	4	4
14	452	14	6	4
15	428	26	8	4
16	576	12	12	4
17	514	4	4	4
18	1072	4	4	4
19	1442	12	4	4
20	2864	16	8	4
21	2534	16	8	4
22	4608	8	8	4
23	6402	8	4	4
24	12448	10	6	4
25	9350	22	6	4
26	-	-	-	4
27	-	-	-	4
28	-	-	-	4
29	-	-	-	4
30	-	4	4	4

Acknowledgments

We would like to thank Saugata Basu, Nikhil Devanur and Tejas Iyer for useful comments and discussions. We also thank the anonymous referees for their comments.

References

[1] A. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. In *AAAI Symposium on Relevance*, 1994.

[2] A. Blum. Open problems. COLT, 2003.

[3] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1993.

[4] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[5] N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC learning of DNF with membership queries under the uniform distribution. In *Annual Conference on Computational Learning Theory*, pages 286–295, 1999.

[6] A. Granville. Arithmetic properties of binomial coefficients. Available at www.dms.umontreal.ca/˜andrew/Binomial.

[7] D. Helmbold, R. Sloan, and M. Warmuth. Learning integer lattices. *SIAM Journal of Computing*, 21(2):240–266, 1992.

[8] J. Jackson. An efficient membership-query algorithm for learning dnf with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[9] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[10] Y. Mansour. An $o(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995.

[11] E. Mossel, R. O'Donnel, and R. Servedio. Learning juntas. In *STOC*, pages 206–212, 2003.

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

[13] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.

[14] K. Verbeurgt. Learning sub-classes of monotone DNF on the uniform distribution. In *Michael M. Richter, Carl H. Smith, Rolf Wiehagen, and Thomas Zeugmann, editors, Algorithmic Learning Theory, 9th International Conference*, pages 385–399, 1998.

[15] J. von zur Gathen and J. Roche. Polynomials with two values. *Combinatorica*, 17(3):345–362, 1997.