

Testing Properties of Boolean Functions:

Lower Bounds on Testing Fourier Degree

Pooya Hatami *

Thesis Advisor: Alexander Razborov

Abstract

We consider the problem of deciding whether a given object has a given property or it is far from any object with the property, referred to as *property testing*. We focus on the case where the objects are Boolean functions, and we survey some of the previously known results about testing for properties such as the number of relevant variables and Fourier degree of a Boolean function. We present the recently developed technique for proving lower bounds in property testing, which is based on showing connections between property testing and communication complexity [13].

For the problem of testing whether a Boolean function has Fourier degree $\leq k$ or it is ϵ -far from any Boolean function with Fourier degree $\leq k$, we improve the known lower bound of $\Omega(k)$ [13, 18], to $\Omega(k/\sqrt{\epsilon})$, using reductions from communication complexity.

*Department of Computer Science, University of Chicago. Email:pooya@cs.uchicago.edu

Contents

1	Introduction	3
2	Motivation	6
2.1	Property Testing, Program Testing and PCP	6
2.2	Property Testing and Learning	7
3	Properties of Boolean Functions	8
3.1	Number of Relevant Variables	8
3.2	Fourier Degree at most d	11
4	Approximate Property Testing	12
5	Lower Bounds Via Communication Complexity	15
5.1	Communication Complexity	15
5.2	Fourier Analysis of Boolean Functions	17
5.3	Reduction from Communication Complexity	18
5.3.1	From Property Testing to Communication Complexity	18
5.3.2	Lower Bounds	19
6	Our Results	21
6.1	Our Constructions of Functions	21
6.2	OR of disjoint copies of DISJ_k^m	24
6.3	Proof of Theorem 1.1	25
6.4	Approximate Fourier degree testing	26

1 Introduction

Property testing is the study of the following type of problems:

Given the ability to perform queries concerning a particular object (e.g., a function, or a graph), the task is to determine whether the object has a predetermined property (e.g. linearity or bipartiteness), or is far from having the property. The task should be performed by inspecting only a small (possibly randomly selected) part of the whole object, where a small probability of failure is allowed.

Notice that this definition is very abstract, and leaves a lot of freedom in how to define a property testing problem. In order to define a property testing problem, we first need to specify what type of queries is the tester allowed to perform. Moreover we need to define a distance measure between the objects, which is required in order to define what it means that the object is *far* from having the property. We assume that the tester is given a *distance* parameter ϵ . The algorithm should **accept** with probability at least $2/3$ every object that has the property, and should **reject** every object that is ϵ -far from any object that has the property.

In case of studying functions, following formulation of Property Testing was suggested in [26]:

Let P be a fixed property of functions, and f be an unknown function. The goal is to determine (possibly probabilistically) if f has property P or if it is far from any function with property P , where distance between functions is measured with respect to some distribution D on the domain of f . More precisely $\text{Dist}(f, g) = \Pr_D(f(x) \neq g(x))$. Towards this end, one is given

examples of the form $(x, f(x))$, where x is distributed according to D . One may also be allowed to query f on instances of one's choice.

The above formulation is inspired by the PAC learning model [40]. In fact, property testing is related to variants of PAC learning as has been shown in [26]. The above formulation allows defining the distance measure according to arbitrary distributions over the domain, it also allows defining property testing problems in which testers observe only randomly chosen instances (rather than being able to query instances of their own choice).

The concept of property testing was introduced in the context of program checking by Blum, Luby and Rubinfeld [16] who showed that *linearity* of a function over a vector space can be tested with a *constant* number of queries. A central ingredient in the proof of the MIP = NEXP theorem [4] was the proof that *multilinearity* can be tested with a *polylogarithmic* number of queries. These two papers were among the roots of the technical developments culminating the PCP Theorem [3, 2].

Rubinfeld and Sudan [37] formally defined property testing in the context of algebraic properties. Subsequently, the interest in property testing was extended to graph properties, with applications to learning and approximation [26]. Over the last two decades, researchers have exerted a considerable amount of effort in testing various properties of a function f , such as whether f is a linear function [16], whether f is isomorphic to a given function [14, 17, 1], whether f is a k -junta [23, 11, 12], a monotone function [21, 24], a dictator [31], a halfspace [29], an s -sparse polynomial, a size- s decision tree, etc. [20] (see, e.g., the survey [34]).

Over the course of this effort, a variety of techniques have been developed for designing property testing algorithms thus proving testing upper bounds. However, as is often

the case in theoretical computer science, lower bounds are harder to come by. Although several lower bounds for known problems are known, until very recently few general techniques were known beyond the use of Yao's minimax lemma. Blais et. al. [13] in a recent paper came up with a new technique to prove property testing lower bounds, by using known lower bounds for randomized communication complexity problems. In particular, they show how to reduce certain communication complexity problems to testing problems, thus showing that communication lower bounds imply lower bounds for property testing. They show that this technique is indeed powerful by applying it on many testing problems and improving on some previous known lower bounds for testing k -linearity, k -juntas, Fourier degree $\leq k$, s -sparse $GF(2)$ -polynomials, etc. All the lower bounds using this technique have been independent of ϵ which is the distance to the property. In Theorem 1.1, we show that this technique can be used to prove a lower bound on testing the property of having lower Fourier degree, which is related to the distance parameter ϵ .

For property testing problems, it is natural to ask what happens to the testing complexity of the problem if we consider relaxations of the original problem. Fischer et al. [23], noticed that the complexity of one of their algorithms for testing k -juntas can be improved to have a quadratic dependence on k if the algorithm is only required to reject functions that are far from being $2k$ -juntas. This relaxation was also considered by Blais et al. [13]. They applied their communication complexity technique to prove a lower bound of $\Omega(\min\{(\frac{k}{t})^2, k\} - \log k)$ on the number of queries necessary for distinguishing between functions that are k -juntas and functions that are ϵ -far from $(k+t)$ -juntas. This of course does not give a good lower bound for $t = \delta k$ when δ is a constant. Ron and Tsur [35] in a recent paper study the problem of testing whether a function is a k -junta or ϵ -far from a $(1 + \delta)k$ -junta. They give a $O(\frac{k \log(1/\delta)}{\epsilon \delta^2})$ upper bound and a $\Omega(k/\log k)$

lower bound for the case when $\epsilon = O(1/\log k)$.

Upper bounds of $2^{O(d)}$ have been given on the general problem of testing whether a Boolean function has Fourier degree $\leq d$ or is ϵ -far from any Boolean function with Fourier degree d [17, 20]. The best lower bounds known on this problem is $\Omega(d)$ [13, 18], which holds for any $\epsilon \leq 1/2$. In this paper we show a lower bound of $\Omega(d/\sqrt{\epsilon})$ for this problem.

Theorem 1.1 (Main Theorem). *Let $\epsilon \geq 2^{-k-1}$. Testing whether a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ has Fourier degree $\leq k$ or ϵ -far from any Boolean function with Fourier degree $\leq k + 1$ requires $\Omega(\frac{k}{\sqrt{\epsilon}})$ queries.*

2 Motivation

The task of testing a property is a relaxation of the task of deciding *exactly* whether an object has the property. Namely, an exact decision procedure is required to accept every object that has the property and reject every object that *does not have* the property. A testing algorithm is promised that the object, either has the property or is far from having the property. While relaxing the task, we expect the algorithm to observe only a small part of the data and to run significantly faster than any exact decision procedure.

2.1 Property Testing, Program Testing and PCP

Property testing of functions was first explicitly defined by Rubinfeld and Sudan [37] in the context of *program testing*. The goal of a program testing algorithm is to test whether a given program computes a specified function. In this context one may choose to test whether the program satisfies a certain property of the specified function, before

checking that it computes the function itself. This approach has been followed in the theory of program testing [16, 37, 36].

Property testing also emerges naturally in the context of probabilistically checkable proofs. In this context the property being tested is whether the function is a codeword of a specific code. This paradigm, explicitly introduced in [5], has shifted from testing codes defined by low-degree polynomials [7, 5, 22, 3, 2] to testing Hadamard codes [2, 9], and to testing the “long code” [27, 38].

2.2 Property Testing and Learning

One of the initial motivations for the study of property testing is its relation to Learning Theory. Assuming that the objects of interest are functions, and instead of being able to query outputs of the function f , at inputs of our choice, we receive a *labeled sample* $\{(x_1, f(x_1)), \dots, (x_m, f(x_m))\}$, where x_i s are random inputs drawn from an unknown distribution D , over the set of possible inputs. In this scenario, the distance between functions is also defined according to the distribution D . This definition of property testing is inspired by the Probably Approximately Correct (PAC) learning model proposed by Valiant [40]. In the PAC model one is given a labeled sample, and the goal is to find a good approximate for f , h . Namely, $\Pr_{x \sim D}[h(x) \neq f(x)] \leq \epsilon$. In the standard PAC model, it is assumed that f belongs to a known class of functions \mathcal{F} , and the algorithm is required to return $h \in \mathcal{F}$ or $h \in \mathcal{H}$ for some class $\mathcal{H} \supseteq \mathcal{F}$.

Therefore, property testing as we defined here is a way of relaxing the PAC learning model, namely, instead of requiring a good approximation function h , we only ask whether such a good approximation *exists* in a given class. One would expect that there exists properties (classes of functions) that property testing is much easier than

PAC learning. This has been shown to be true for example for linear functions [16], multivariate polynomials [37]. and

3 Properties of Boolean Functions

In this section we present some examples of Properties and upper and lower bounds which are known for them.

3.1 Number of Relevant Variables

It is a natural question, when we have a function f on n variables, to ask how many of the variables does f depend on.

Definition 3.1. *A Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is called a k -junta, if there are at most k indices i such that:*

$$\exists x \in \{-1, 1\}^n : f(x) \neq f(x \oplus e_i),$$

where e_i is the vector which is equal to 1 everywhere and is equal to -1 on the i th entry, and the operation \oplus acts as entry-wise multiplication.

In the framework of property testing, one seeks to determine the minimum number of queries to a function required to distinguish k -juntas from functions that are far from being k -juntas. The first result explicitly related to testing juntas was obtained by Parnas, Ron, and Samorodnitsky [31], who generalized a result of Bellare, Goldreich, and Sudan [10] on testing long codes to obtain an algorithm for testing 1-juntas (i.e. dictators) with only $O(1/\epsilon)$ queries.

Soon afterwards, Fischer *et al.* [23] introduced algorithms for testing k -juntas with $\tilde{O}(k^2)/\epsilon$ queries. The original analysis of Fischer *et al.* only applied to functions with a

Boolean range. Diakonikolas *et al.* [20] extended the analysis to handle functions with arbitrary finite ranges.

The algorithm of Fischer *et al.* for testing juntas remained the most query efficient ways to test juntas until Eric Blais [12] presented a new algorithm for testing juntas that significantly improved the previous known upper bounds. Namely, they gave a junta-testing algorithm that required $O(k/\epsilon + k \log k)$ queries. This upper bound almost closes the gap from the best known lower bound for this problem.

The first non-trivial lower bound on the query complexity of the testing juntas problem was provided by Fischer *et al.* [23], who showed that $\Omega(\log k)$ queries are necessary to test k -juntas. The lower bound was improved to $\Omega(k)$ by Chockler and Gutfreund [19]. In a recent paper of Blais *et al.* [13] the $\Omega(k)$ lower bound was proved via a simpler proof, where they present a new method of obtaining property testing lower bounds using *communication complexity* lower bounds. We will present their proof in Section 5.3.

Upper Bound

Here we present the k -Junta test introduced by Eric Blais [12]. This test is based on the simple but useful idea, due to Blum, Hellerstein, and Littlestone [15]: if we have two inputs $x, y \in \mathcal{X}$ such that $f(x) \neq f(y)$, then the set S of coordinates in which x and y disagree contains at least one coordinate that is relevant in f . Furthermore, by performing a binary search over the hybrid inputs formed from x and y , we can identify the relevant coordinate using $O(\log |S|)$ queries.

The idea behind the algorithm is to partition the set of variables to s sets, and trying to find sets which are relevant to the function. Once we have found more than k parts that are relevant to the function we reject, and otherwise we accept.

Definition 3.2. For any $x \in \{-1, 1\}^n$ and a subset $S \subseteq [n]$ denote by x_S the vector $(a_1, \dots, a_{|S|}) \in \{-1, 1\}^{|S|}$ obtained by restricting x to indices in S .

Algorithm 1 (Eric Blais [12]). *k*-JUNTA TEST (f, ϵ).

Additional parameters: $s = O(k^9/\epsilon^5)$, $r = O(k \log k/\epsilon)$, $S \rightarrow \emptyset$

1. Randomly partition the coordinates in $[n]$ into s sets I_1, \dots, I_s .

2. For each of r rounds,

2.1 Generate a pair (x, y) at random such that $x_S = y_S$.

2.2 If $f(x) \neq f(y)$, then do a binary search to identify a set I_j with a relevant variable and add I_j to S .

2.3 If S contains greater than k sets, quit and **reject**.

3. **Accept**.

This algorithm obviously accepts any k -Junta, with probability 1. The key to showing that functions ϵ -far from k -juntas are rejected with high probability relies on the next lemma which proves a lower bound on the influence of union of k parts in a random partition of a k -junta.

Definition 3.3. The influence of the set $S \subseteq [n]$ in a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is defined to be

$$\text{Inf}_f(S) = 2\Pr_{x, y \in \{0, 1\}^n} [f(x) \neq f(x_S y_{\bar{S}})].$$

Lemma 3.4 ([12]). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be ϵ -far from being a k -junta, and let \mathcal{I} be a sufficiently fine partition of $[n]$. Then with high probability, for any set S formed by taking the union of at most k parts of \mathcal{I} , $\text{Inf}_f(\bar{S}) \geq \epsilon/2$.

Assume that f is ϵ -far from being a k -junta and at most k relevant parts have been identified by the algorithm. Lemma 3.4 implies that influence of the remaining variables is at most $\epsilon/2$. Thus with probability at least $\epsilon/2$, for a random pair (x, y) sampled in step 2.1, $f(x) \neq f(y)$.

3.2 Fourier Degree at most d

In this section we present known results for the problem of testing whether a given Boolean function has low Fourier degree. For convenience, in the context of Fourier analysis we consider the Boolean function to be of the form $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$.

Definition 3.5. *It is a well known fact that for the set of characters $\chi_S = \prod_{i \in S} x_i$, every Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ has a unique representation in the form*

$$f = \sum_{S \subseteq [n]} \chi_S \hat{f}(S),$$

where $\hat{f}(S) \in \mathbb{R}$ are called the Fourier coefficients of f . The Fourier degree of a Boolean function f is equal to the maximum $k \geq 0$ such that $\hat{f}(S) \neq 0$ for a set S of size k .

Diakonikolas *et al.* [20] considered the problem of testing whether a Boolean function f has Fourier degree at most d . They proved a general lower bound of $\tilde{\Omega}(\log d)$, and a lower bound of $\tilde{\Omega}(\sqrt{d})$ for the non-adaptive tester and any $\epsilon \leq 1/2$. They also present an algorithm with $\tilde{O}(2^{6d}/\epsilon^2)$ query complexity for this problem. Chakraborty *et al.* [18] and later Blais *et al.* [13] improved the lower bound to $\Omega(d)$, for any $0 \leq \epsilon \leq 1/2$. We present the lower bound given by Blais *et al.* in Section 5.3.

4 Approximate Property Testing

A natural question, which was raised in [23] is whether the testing problems become easier tasks if we relax the soundness requirement. That is for example in the case of testing for number of relevant variables, we only require that the test accepts every k -junta, and rejects inputs which are ϵ -far from any $2k$ -junta.

This type of relaxation was later considered by Blais *et al.* [13], where they used their new technique to prove a lower bound of $\Omega(\min\{(k/t)^2, k\} - \log k)$ on the number of queries necessary for distinguishing between functions that are k -junta and functions that are ϵ -far from any $(k+t)$ -junta. Ron and Tsur [35] studied the problem of testing whether a function is k -junta or ϵ -far from any $(1+\gamma)k$ -junta, where γ is a constant. They prove that even constant γ does not make the junta testing problem much easier.

Theorem 4.1 (Ron, Tsur [35]). *Any algorithm that distinguishes between the case that f is a k -junta and the case that f is ϵ -far from any $(1+\gamma)k$ -junta for constant ϵ and γ must perform $\Omega(k/\log(k))$ queries.*

Moreover they show slight improvements on the upper bound for the relaxed junta testing problem.

Theorem 4.2 (Ron, Tsur [35]). *There exists an algorithm that, given query access to $f : \{0,1\}^n \rightarrow \{0,1\}$ and parameters $k \geq 1$, and $0 < \epsilon, \gamma < 1$, distinguishes with high constant probability between the case that f is a k -junta and the case that f is ϵ -far from any $(1+\gamma)k$ -junta. The algorithm performs $O\left(\frac{k \log(1/\gamma)}{\epsilon \gamma^2}\right)$ queries.*

Here we present their proof of the lower bound.

Proof of Theorem 4.1. The lower bound is established by a reduction from the *Distinct Elements* problem.

Distinct Elements Problem: An algorithm is given query access to a string s and must compute approximately and with high probability the number of distinct elements appearing in s .

For a string of length t , this problem is equivalent to approximating the support size of a distribution where the probability for every event is in multiples of $1/t$ [32]. Valiant and Valiant prove that

Theorem 4.3 (Valiant and Valiant [39], rephrased). *For any constant $\phi > 0$, there exists a pair of distributions p^+ and p^- for which each domain element occurs with probability at least $1/t$, satisfying:*

1. $|S(p^+) - S(p^-)| = \phi \cdot t$, where $S(D) = |\{x : \Pr_D[x] > 0\}|$.
2. Any algorithm that distinguishes p^+ from p^- with probability at least $2/3$ must obtain $\Omega(t/\log t)$ samples.

A simple, corollary of the above theorem is a lowerbound of $\Omega(t/\log t)$ for the task of distinguishing between a string of length t with $t/2$ distinct elements from one with fewer than $t/16$ distinct elements. The proof follows by presenting a reduction from this problem.

Assume $k = n/8$. Here we will present a mapping from strings of length $t = \Theta(n)$ to functions from $\{0, 1\}^n$ to $\{0, 1\}$, with the property that: An algorithm that with high probability distinguishes between k -juntas and functions that are far from any $2k$ -junta using q -queries, can be used to distinguish between strings with at most $k - \Theta(\log(k))$ colors and strings with at least $8k - \Theta(\log k)$ colors using q queries.

Let F_m^n be the family of Boolean functions from $\{0, 1\}^n$ to $\{0, 1\}$, such that for each $U \subseteq \{\log(n) + 1, \dots, n\}$ of size m , and a surjective function $\psi : \{0, 1\}^{\log(n)} \rightarrow U$, there is

a function $f^{U,\psi}$ in F_m^n defined as

$$f^{U,\psi}(x_1, \dots, x_n) = x_{\psi(x_1, \dots, x_{\log(n)})}.$$

It is easy to see that functions in F_m^n depend on $\log(n) + m$ variables. Following claim is the main fact that the reduction relies on.

Claim 4.4 ([35]). *For any constant value c and for $t > n/c$, every function in $F_{t/2}^n$ is ϵ -far from any $t/4$ -junta, for a constant $\epsilon > 0$.*

Now we are ready to explain the reduction. Let $s = s_1 s_2 \dots s_n$ be a string of length n , where $s_i \in \{1, \dots, n - \log(n)\}$. Also let $b : \{0, 1\}^{\log(n)} \rightarrow \{0, \dots, n - 1\}$ be the function that maps each binary representation to the corresponding number in $\{0, \dots, n - 1\}$. Define $f_s : \{0, 1\}^n \rightarrow \{0, 1\}$ as follows

$$f_s(x_1, \dots, x_n) = x_{(\log(n) + s_b(x_1, \dots, x_{\log(n)}))}.$$

Following claim is easy to see

Claim 4.5 ([35]). *The following are true about the reduction f_s*

1. *For a string s , each query to the function f_s of the form $f(x_1, \dots, x_n)$ can be answered by performing a single query to s .*
2. *For a string s with $n/2$ colors the function f_s belongs to $F_{n/2}^n$.*
3. *For a string s with $n/16$ colors the function f_s belongs to $F_{n/16}^n$.*

Which completes the proof. Notice that we had assumed that $k = n/8$. The case $k < n/8$ can be handled by a padding argument.

□

One could consider similar relaxations of other property testing problems. In the case of testing Fourier degree of Boolean functions, the known lower bounds hold for relaxations of the Fourier degree testing as well. Since in this case, the gap between the best known upper and lower bound is exponential, it is natural to ask whether it is possible to improve the $O(2^{O(k)})$ upper bound for approximate version of Fourier degree testing, namely, given k and a constant $c > 0$, deciding whether a Boolean function has Fourier degree at most k or it is ϵ -far from any Boolean function with Fourier degree at most ck .

5 Lower Bounds Via Communication Complexity

5.1 Communication Complexity

In this section we will give a brief introduction to Communication Complexity, and state known lower bounds for the famous set disjointness problem. The two-party communication model was introduced by Andrew Chi-Chih Yao [41] in 1979. In this model, two parties, traditionally called Alice and Bob, are trying to collaboratively compute a known Boolean function $F : X \times Y \rightarrow \{0, 1\}$. Each party is computationally unbounded; however, Alice is only given input $x \in X$ and Bob is only given $y \in Y$. In order to compute $F(x, y)$, Alice and Bob communicate in accordance with an agreed-upon *communication protocol* \mathcal{P} . Protocol \mathcal{P} specifies as a function of transmitted bits only whether the communication is over and, if not, who sends the next bit. Moreover, \mathcal{P} specifies as a function of the transmitted bits and x the value of the next bit to be sent by Alice. Similarly for Bob. The communication is over as soon as one of the parties knows the value of $F(x, y)$. The cost of the protocol \mathcal{P} is the number of bits exchanged

on the worst input.

Definition 5.1. *The deterministic communication complexity of F , denoted by $DC(F)$, is the cost of an optimal communication protocol computing F .*

There are several ways in which the deterministic communication model can be extended to include randomization. In the *public-coin* model, Alice and Bob have access to a shared random string r chosen according to some probability distribution. The only difference in the definition of a protocol is that now the protocol \mathcal{P} specifies the next bit to be sent by Alice as a function of x , the already transmitted bits, and a random string r . Similarly for Bob. In the *private-coin* model, Alice has access to a random string r_A hidden from Bob, and Bob has access to a random string r_B hidden from Alice.

Definition 5.2. *The bounded-error randomized communication complexity of F with public coins (private coins), denoted by $RC_2(F)$ ($RC_2^{\text{pri}}(F)$), is the minimum cost of a public-coin (private-coin) randomized protocol that computes F correctly with probability at least $2/3$ on every input. (The subscript 2 refers to permitting 2-sided error.)*

Clearly, for every Boolean F we have $RC_2(F) \leq RC_2^{\text{pri}}(F)$. Ilan Newman [30] showed that the two measures are identical up to constant multiplicative factors and logarithmic additive terms.

Theorem 5.3 (Newman [30]). *For every Boolean function $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ we have*

$$RC_2^{\text{pri}}(F) = O(RC_2 + \log n).$$

Set Disjointness

Alice and Bob are given x and y , $x, y \in \{-1, 1\}^n$, and compute

$$\text{DISJ}(x, y) = \bigvee_{i=1}^n (x_i \wedge y_i),$$

where $(a \wedge b) = 1$ if $a = b = 1$, and -1 otherwise.

It is well-known that $\text{RC}_2(\text{DISJ}^n) = \Omega(n)$. The problem DISJ_k^n is a balanced version of DISJ^n with the promise that $|x| = |y| = k$, where $|x|$ is equal to the number of 1s in x , and that $x_i \wedge y_i = 1$ for at most one i . A first lower bound of $\Omega(\sqrt{n})$ when $k = n/3$ was proved by Babai et al. [6]. This bound was strengthened to $\Omega(k)$, by Kalyanasundaram and Schnitger [28], simplified by Razborov [33], and further simplified by Bar-Yossef et al. [8].

5.2 Fourier Analysis of Boolean Functions

Consider the 2^n -dimensional vector space of all functions $f : \{-1, 1\}^n \rightarrow \mathbb{R}$. An inner product on this space can be defined as follows

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{-1, 1\}^n} f(x)g(x) = \mathbf{E}[f \cdot g],$$

where the latter expectation is taken uniformly over all $x \in \{-1, 1\}^n$. This defines the l_2 -norm

$$\|f\|_2 = \sqrt{\langle f, f \rangle} = \sqrt{\mathbf{E}[f^2]}.$$

Definition 5.4. For $S \subseteq [n]$, the character $\chi_S : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is defined as

$$\chi_S(x) = \prod_{i \in S} x_i.$$

The set of characters forms an orthonormal basis for the inner product space. Hence, every function $f : \{-1, 1\}^n \rightarrow \mathbb{R}$ can be written uniquely as

$$f = \sum_S \langle f, \chi_S \rangle \chi_S.$$

The above equation is referred to as the Fourier expansion of f , and the Fourier coefficient of f corresponding to set S is defined as

$$\widehat{f}(S) = \langle f, \chi_S \rangle.$$

Definition 5.5. *The Fourier degree of a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is equal to maximum $k > 0$ such that there exists $S \subseteq [n]$, $|S| = k$, for which $\widehat{f}(S) \neq 0$.*

5.3 Reduction from Communication Complexity

In this section we present the approach which was introduced recently by Blais *et al.* [13], to prove property testing lower bounds via communication complexity. Blais *et al.* give a simple scheme for reducing communication problems to testing problems, which allows them to use known lower bounds in communication complexity to prove lower bounds in testing. In this subsection, we will present proofs by Blais *et al.* for lower bounds on testing k -linear functions, k -juntas, and functions of Fourier degree at most k .

5.3.1 From Property Testing to Communication Complexity

In this section we present the general reduction introduced by Blais *et al.*, in a slightly different way. Namely, the reduction from a communication complexity problem, sometimes helps prove lower bound for testing not a property, but a property testing problem, where we want to decide where the function belongs to property F , or property G . For two families of functions \mathcal{F} and \mathcal{G} , let $P(\mathcal{F}, \mathcal{G})$ be the testing problem of determining whether a given function belongs to \mathcal{F} or \mathcal{G} . Let us denote by $Q(P)$, the query complexity of a property testing problem P , where the tester is allowed to make two-sided error up to $1/3$.

Given a testing problem $P(\mathcal{F}, \mathcal{G})$, Boolean functions $f, g : \{-1, 1\}^n \rightarrow \{-1, 1\}$, and a “combining” function $h = h(f, g)$, define the following communication game $C_{h,P}$:

Alice knows f and Bob knows g , and their goal is to decide if h belongs to \mathcal{F} or it is in \mathcal{G} .

Lemma 5.6 ([13], rephrased and partially stated). *For any function h , properties \mathcal{F} and \mathcal{G} ,*

1. $\text{RC}(C_{h,P(\mathcal{F},\mathcal{G})}) \leq 2Q(P(\mathcal{F},\mathcal{G})),$

2. $\text{RC}^1(C_{h,P(\mathcal{F},\mathcal{G})}) \leq 2Q^1(P(\mathcal{F},\mathcal{G})),$

where RC^1 and Q^1 are one sided error complexities.

Proof. The proof follows by showing how to use a t -query testing algorithm that distinguishes between \mathcal{F} and \mathcal{G} to create a communication protocol for $C_{h,P(\mathcal{F},\mathcal{G})}$. Alice and Bob can use public randomness to generate the required queries of the testing algorithm. For a query to $h(x)$, Alice can compute $f(x)$, and Bob can compute $g(y)$. Now they can communicate $f(x)$ and $g(y)$, and Bob can compute $h(f(x),g(x))$ and answer to the query of the tester with the value $h(f(x),g(y))$. After t queries, there has been $2t$ bits of communication, and they can use the decision of the tester on whether h has property \mathcal{F} or it has property \mathcal{G} .

The proof of (2) is analogous. □

5.3.2 Lower Bounds

Definition 5.7. *The Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is k -linear, i.e. a parity function on k variables, when there is a set $S \subseteq [n]$, $|S| = k$, such that for every $x \in \{-1, 1\}^n$, $f(x) = \prod_{i \in S} x_i$.*

The problem of testing k -linear functions was first studied by Fischer *et al.* [23]. Goldreich [25] proved an $\Omega(\sqrt{k})$ lower bound for testing k -linear functions. He also proved that $\Omega(k)$ queries are required to non-adaptively testing k -linear functions, and conjectured that this lower bound holds for all testers. Blais *et al.* [13] and in independent work Eric Blais and Daniel Kane confirmed Goldreich's conjecture.

Theorem 5.8 ([13], partially stated). *Fix $1 < k < n - 1$. Then $\Omega(\min\{k, n - k\})$ queries are required to test*

- (i) *k -linear functions,*
- (ii) *k -juntas,*
- (iii) *Boolean functions of Fourier degree at most k .*

Following proposition is the key to the proof of Theorem 5.8.

Proposition 5.9. *A $(k + 2)$ -linear function is $\frac{1}{2}$ -far from*

- *k -linear functions,*
- *k -juntas, and*
- *Boolean functions of Fourier degree at most k .*

Proof. Since all k -linear functions, and all k -juntas have Fourier degree at most k , we just need to prove the statement for Boolean functions of Fourier degree at most k . Let χ_U , be a $(k + 2)$ -linear function for some $U \subseteq [n]$, $|U| = k + 2$. Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a Boolean function of Fourier degree at most k , we can write $f = \sum_{S \subseteq [n], |S| \leq k} \widehat{f}(S) \chi_S$. The distance between χ_U and f is equal to $\frac{1}{2}(1 - \langle f, \chi_U \rangle)$, and

$$\langle f, \chi_U \rangle = \sum_{S \subseteq [n], |S| \leq k} \widehat{f}(S) \cdot \langle \chi_S, \chi_U \rangle = 0,$$

implying that the distance between f and χ_U is $1/2$. □

Proof of Theorem 5.8. First we will present the proof for the case $k \leq n/4$.

Let k be even, and choose $k' = k/2 + 1$. Let \mathcal{F} be the family of k -linear functions, and \mathcal{G} be the family of $k + 2$ -linear functions. Let f be the parity function on the set of

variables where $x_i = 1$, and g be the parity function on the set of variables where $y_i = 1$, where x is the input to Alice and y is the input to Bob. Let $h = h(f, g) := f \cdot g$. Also let $C_{h, P(\mathcal{F}, \mathcal{G})}$ have the extra promise that $|x| = |y| = k'$, and $x_i = y_i = 1$ for at most one i .

The definition of $C_{h, \mathcal{P}}$ coincides with the communication problem DISJ_k , thus by Lemma 5.6 we have

$$2Q(P(\mathcal{F}, \mathcal{G})) \geq \text{RC}(C_{h, P(\mathcal{F}, \mathcal{G})}) = \text{RC}(\text{DISJ}_k) = \Omega(k).$$

The theorem now follows from the above inequality and Proposition 5.9. The case where $k > 3n/4$ can be handled by multiplying f and g by the parity function $\chi_{[n]}$, and the case where $n/4 < k < 3n/4$ can be proved by a simple padding argument. \square

6 Our Results

6.1 Our Constructions of Functions

In this section we give a method how to construct functions which are of Fourier degree $\leq k$ and how to construct functions which are far from having Fourier degree at most k .

Let l be a positive integer. For any $S \subseteq [n]$, $|S| = l$, let $C_{a_1, a_2, \dots, a_l}^S$ for any $(a_1, \dots, a_l) \in \{-1, 1\}^l$ be a subset of $[n] \setminus S$.

Now for any $S \subseteq [n]$ and sets $\{C_{a_1, \dots, a_l}^S\}$, let $f^S : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be the Boolean function defined as following

$$f^S(x_1, \dots, x_n) = \chi_{C_{x_S}}(x_1, \dots, x_n),$$

where $\chi_A(x_1, \dots, x_n) = \prod_{i \in A} x_i$ for $A \subseteq [n]$.

In the next two propositions we show how cardinalities of sets C_{a_1, \dots, a_l} can lead f^S to be of low Fourier degree, or to be far from any Boolean function with Fourier degree k .

Proposition 6.1. *The Boolean function $f^{[l]} : \{-1, 1\}^n \rightarrow \{-1, 1\}$ described above, is of Fourier degree $m + l$ if*

$$\forall (a_1, \dots, a_l) \in \{-1, 1\}^l, |C_{a_1, \dots, a_l}^{[l]}| \leq m,$$

where $[l] = \{1, 2, \dots, l\}$.

Proof.

For the sake of simplicity we will not write the superscript $[l]$ for $C_{a_1, \dots, a_l}^{[l]}$. We have to prove that $\langle f, \chi_S \rangle = 0$ for any $S \subseteq [n]$ with $|S| \geq m + l + 1$.

$$\begin{aligned} \widehat{f}^{[l]}(S) &= \langle f^{[l]}, \chi_S \rangle = 2^{-n} \sum_{x \in \{-1, 1\}^n} \chi_{C_{x_{[l]}}}(x) \cdot \chi_S(x) \\ &= 2^{-n} \sum_{x_1, \dots, x_l} \sum_{x_{l+1}, \dots, x_n} \chi_{C_{x_1, \dots, x_l}}(x_1, \dots, x_n) \cdot \chi_S(x_1, \dots, x_n) = 0. \end{aligned}$$

The last equality follows from the fact that

$$\sum_{x_{l+1}, \dots, x_n \in \{-1, 1\}} \chi_{C_{x_1, \dots, x_l}}(x_1, \dots, x_n) \cdot \chi_S(x_1, \dots, x_n) = 0,$$

since

$$\exists i \in S : i \notin C_{x_1, \dots, x_l} \cup \{1, \dots, l\},$$

because $|S| \geq m + l + 1$. □

Proposition 6.2. *The Boolean function $f^{[l]}$ is $1/2^{l+1}$ -far from any Boolean function of Fourier degree $m - 1$ if for only one $(b_1, \dots, b_l) \in \{-1, 1\}^l$, $|C_{(b_1, \dots, b_l)}| \geq m$, and*

$$\forall (a_1, \dots, a_l) \neq (b_1, \dots, b_l) : |C_{a_1, \dots, a_l}| \leq m - 1.$$

Proof. First we prove that for any $U \subseteq \{1, \dots, l\}$, the fourier coefficient of $|f^{[l]}|$ at $S = U \cup C_{b_1, \dots, b_l}$ is equal to $1/2^l$.

$$\begin{aligned}
\widehat{f}^{[l]}(S) &= \langle f^{[l]}, \chi_{U \cup C_{b_1, \dots, b_l}} \rangle \\
&= 2^{-n} \sum_{x \in \{-1, 1\}^n} \chi_{x_{[l]}}(x) \cdot \chi_{U \cup C_{b_1, \dots, b_l}}(x) \\
&= 2^{-n} \sum_{x_{[l]} \in \{-1, 1\}^l} \left(\prod_{i \in U} x_i \right) \sum_{x_{l+1}, \dots, x_n} \chi_{x_{[l]}}(x) \chi_{C_{b_1, \dots, b_l}}(x) \\
&= 2^{-l} \prod_{i \in U} b_i \\
&\quad + 2^{-n} \sum_{x_{[l]} \neq (b_1, \dots, b_l)} \left(\prod_{i \in U} x_i \right) \sum_{x_{l+1}, \dots, x_n} \chi_{x_{[l]}}(x) \cdot \chi_{C_{b_1, \dots, b_l}}(x) \\
&= 2^{-l} \prod_{i \in U} b_i.
\end{aligned}$$

The last equality follows from the fact that if $(a_1, \dots, a_l) \neq (b_1, \dots, b_l)$ then $|C_{(b_1, \dots, b_l)}| > |C_{(a_1, \dots, a_l)}|$, therefore

$$\sum_{x_{l+1}, \dots, x_n} \chi_{x_{[l]}}(x) \cdot \chi_{C_{b_1, \dots, b_l}}(x) = 0.$$

Let $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a Boolean function with fourier degree $m - 1$, thus we can write

$$g(x) = \sum_{S \subseteq [n]}^{|S| \leq m-1} \widehat{g}(S) \chi_S(x).$$

Notice that the distance between two functions f and g with range $\{-1, 1\}$ can be written as $\frac{1}{2} \|f - g\|_2^2 = \frac{1}{2} \mathbf{E}[(f - g)^2]$. Finally Parseval's identity implies that

$$\begin{aligned}
\|f - g\|_2^2 &= \sum_{S \subseteq [n]}^{|S| \leq m-1} (\widehat{f}(S) - \widehat{g}(S))^2 + \sum_{S \subseteq [n]}^{|S| \geq m} \widehat{f}(S)^2 \\
&\geq \sum_{S \subseteq [n]}^{|S| \geq m} \widehat{f}(S)^2 \geq \sum_{U \subseteq [l]} \left(2^{-l} \prod_{i \in U} b_i \right)^2 = 2^{-l}.
\end{aligned}$$

□

Proposition 6.3. *The Boolean function $f^{[l]}$ is $1/2^{2l+1}$ -far from any Boolean function of Fourier degree $m + l - 1$ if for only one $(b_1, \dots, b_l) \in \{-1, 1\}^l$, $|C_{b_1, \dots, b_l}| \geq m$, and*

$$\forall (a_1, \dots, a_l) \neq (b_1, \dots, b_l) : |C_{(a_1, \dots, a_l)}| \leq m - 1.$$

Proof. The proof is similar to the proof of Proposition 6.2, with the difference that we only use the fact that $\widehat{f^{[l]}}(\{1, \dots, l\} \cup C_{b_1, \dots, b_l}) = 2^{-l}$, and thus f is 2^{-2l-1} far from any function with fourier degree $m + l - 1$. □

6.2 OR of disjoint copies of DISJ_k^m

In this section we present a new communication problem which later will be used to prove our lower bounds.

$\text{DISJ}_k^{l,m}$: Alice and Bob are given Boolean strings of length lm , where $x, y \subseteq \{-1, 1\}^{lm}$, with the extra promise that for every i ,

$$|x_{\{(i-1)m+1, \dots, im\}}| = |y_{\{(i-1)m+1, \dots, im\}}| = k,$$

and $x_j \wedge y_j = 1$ for at most one $j \in \{(i-1)m+1, \dots, im\}$. The goal is to compute

$$\bigvee_i \text{DISJ}_k(x_{\{(i-1)m+1, \dots, im\}}, y_{\{(i-1)m+1, \dots, im\}}).$$

$\text{DISJ}_k^{l,m}$ is basically OR of l disjoint copies of DISJ_k^m .

Lemma 6.4. $\text{RC}(\text{DISJ}_k^{l,m}) = \Omega(lk)$.

Proof. We reduce $\text{DISJ}_{lk/4}^{lk}$ to $\text{DISJ}_k^{l,m}$. The reduction is very simple. Assume there is a protocol \mathcal{P} for solving $\text{DISJ}_k^{l,m}$. Alice and Bob are given strings x and y respectively, of length lk , and want to decide whether they are disjoint or not. They are also promised that $x_i = y_i = 1$ for at most one choice of i . Let both Alice and Bob divide x and y to l strings of length k , $x = x^1x^2\dots x^l$ and $y = y^1y^2\dots y^l$. Now for each i , Alice constructs

$$x_c^i = 1^{k-|x^i|}(-1)^{m-2k+|x^i|},$$

where π^a represents concatenation of a copies of π . Bob constructs

$$y_c^i = (-1)^{m-2k+|y^i|}1^{k-|y^i|}.$$

Finally they use protocol \mathcal{P} to solve $\text{DISJ}_k^{l,m}$ on inputs

$$x' = x^1x_c^1x^2x_c^2\dots x^lx_c^l \quad \text{and} \quad y' = y^1y_c^1y^2y_c^2\dots y^ly_c^l.$$

Notice that the choice of x_c^i and y_c^i makes every block in x' and every block in y' have exactly k , 1s. Moreover, this construction preserves the property that there is only one i for which $x_i' = y_i' = 1$, and the resulting problem is equivalent to the original disjointness problem.

As a result

$$\text{RC}(\text{DISJ}_k^{l,m}) \geq \text{RC}(\text{DISJ}_{lk/4}^{lk}) = \Omega(lk).$$

□

6.3 Proof of Theorem 1.1

In this section, we show how to use the communication complexity technique to prove a lower bound of $\Omega(k/\sqrt{\epsilon})$ on testing whether a Boolean function is of Fourier degree at most k .

Proof of Theorem 1.1. Let l be the largest integer such that $\epsilon < 2^{-2^{l-1}}$. We prove that $\Omega(k \cdot 2^l)$ queries are required to test whether a Boolean function has Fourier degree $\leq k$ or is ϵ -far from any Boolean function with degree $\leq k + 1$. Notice that since $\epsilon \geq 2^{-k-1}$ thus $l \leq \frac{k}{2}$.

Let $f^{[l]} : \{-1, 1\}^n \rightarrow \{-1, 1\}$, be defined for a family of subsets of $[n]$, $\{C_{a_1, \dots, a_l}^{[l]}\}$ as explained in Section 6.1. Similarly define $g^{[l]} : \{-1, 1\}^n \rightarrow \{-1, 1\}$ for a family of sets $\{D_{a_1, \dots, a_l}^{[l]}\}$. Let $h = h(f, g) = f \cdot g$. Alice is given f and Bob is given g , and their goal, (testing problem P), is to decide whether $h(f, g)$ has Fourier degree at most k or it is ϵ -far from every Boolean function with Fourier degree at most k .

Assume that $k - l$ is even, and let $C_{h, P}$ have the extra promise that for every $(a_1, \dots, a_l) \in \{-1, 1\}^l$, $|C_{a_1, \dots, a_l}^{[l]}| = |D_{a_1, \dots, a_l}^{[l]}| = (k - l)/2 + 1$. Moreover with the promise that $|C_{a_1, \dots, a_l}^{[l]} \cap D_{a_1, \dots, a_l}^{[l]}| \leq 1$.

Notice that $C_{h, P}$ is equivalent to $\text{DISJ}_{(k-l)/2+1}^{2^l, n-k}$, where the i th block of the input to Alice represents $C_{a_1, \dots, a_l}^{[l]}$ and i th block of the input to Bob represents $D_{a_1, \dots, a_l}^{[l]}$, where (a_1, \dots, a_l) is the i -th vector in $\{-1, 1\}^l$ in the chronological order. By Lemma 5.6 we have

$$2Q(h) \geq \text{RC}(C_{h, P}) = \text{DISJ}_{(k-l)/2+1}^{2^l, n-k} = \Omega(k \cdot 2^l),$$

where the last equality follows from Lemma 6.4. Now the result follows by Proposition 6.1 and Proposition 6.2. \square

6.4 Approximate Fourier degree testing

Chakraborty *et al.* [18] proved that testing whether a Boolean function has Fourier degree at most k or it is far from any Boolean function with Fourier degree $n - \Theta(1)$ requires

$\Omega(k)$ queries. Here we prove an $\Omega(1/\epsilon)$ lower bound for the non-adaptive tester, using Yao's minimax principle. For this we introduce two distributions D_p and D_n where D_p is a distribution restricted to a subset of Boolean functions with Fourier degree $\leq k$ and D_n is a distribution restricted to a subset of Boolean functions ϵ -far from any Boolean function with Fourier degree $\leq n - 2k$. Which combined with Chakraborty *et al.*'s result gives an $\Omega(k + 1/\epsilon)$ lower bound for non-adaptively approximate testing the Fourier degree.

Theorem 6.5. *Let $\epsilon \geq 1/2^{-k/2-1}$. Non-adaptively Testing whether a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ has Fourier degree $\leq k$ or it is ϵ -far from any Boolean function with Fourier degree $\leq n - k$ requires $\Omega(\frac{1}{\epsilon} + k)$ queries.*

Proof. Let l be the largest integer such that $\epsilon < 2^{-l-1}$. We prove that $\Omega(2^l)$ queries are required to test whether a Boolean function has Fourier degree $\leq k$ or is ϵ -far from any Boolean function with degree $\leq n - k$. Notice that since $\epsilon \geq 1/2^{\frac{k}{2}-1}$ thus $l \leq \frac{k}{2} - 1$.

Let D_p be the distribution where for any $(a_1, \dots, a_l) \in \{-1, 1\}^l$ we choose uniformly at random $C_{(a_1, \dots, a_l)}$ to be a subset of size $k/2$ of $\{l + 1, \dots, n\}$. Finally constructing $f^{[l]}$ using the chosen sets as described in Section 6.1. Proposition 6.1 immediately implies that $f^{[l]}$ has Fourier degree $\leq k$.

Let D_n be the distribution where we choose $(b_1, \dots, b_l) \in \{-1, 1\}^l$ uniformly at random and choose $C_{(b_1, \dots, b_l)}$ to be a subset of cardinality $n - k + 1$ of $\{l + 1, \dots, n\}$. Also for any $(a_1, \dots, a_l) \in \{-1, 1\}^l$, where $(a_1, \dots, a_l) \neq (b_1, \dots, b_l)$, we choose uniformly at random $C_{(a_1, \dots, a_l)}$ to be a subset of cardinality $k/2$ of $\{l + 1, \dots, n\}$. Finally build $f^{[l]}$ using the chosen sets. Proposition 6.2 immediately implies that $f^{[l]}$ is 2^{-l-1} -far from any Boolean function with Fourier degree $\leq n - k$.

Let our final distribution be that with probability $1/2$ we draw $f^{[l]}$ from D_p and with

probability $1/2$ we draw $f^{[l]}$ from D_n . Now by Yao's minimax principle if we prove that any deterministic algorithm that queries less than $2^l/6$ with constant probability makes a mistake, implies that the original testing problem with constant probability of error requires $\frac{2^l}{6} = \Omega(\frac{1}{\epsilon})$ queries.

For any deterministic set of $d \leq \frac{2^l}{6}$ queries to outputs of the function on inputs x^1, \dots, x^d ,

$$|\{(a_1, \dots, a_l) | (\exists 1 \leq i \leq d) x_{[l]}^i = (a_1, \dots, a_l)\}| \leq d \leq \frac{2^{l-1}}{6}.$$

Therefore the measure of the set of functions from support of D_n for which the deterministic tester has not yet queried any input from the high degree subcube is at least

$$\frac{1}{2} \cdot \frac{2^l - 2^l/6}{2^l} = \frac{5}{12} \geq \frac{1}{3}.$$

Thus with probability at least $\frac{1}{3}$ the deterministic tester will make an error. \square

References

- [1] Noga Alon and Eric Blais. Testing boolean function isomorphism. In *APPROX-RANDOM'10*, pages 394–405, 2010.
- [2] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45:501–555, May 1998.
- [3] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, 1998.
- [4] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. In *Proceedings of the 31st Annual Symposium on Foun-*

- dations of Computer Science*, SFCS '90, pages 16–25 vol.1, Washington, DC, USA, 1990. IEEE Computer Society.
- [5] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, STOC '91, pages 21–32, New York, NY, USA, 1991. ACM.
- [6] Laszlo Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 337–347, Washington, DC, USA, 1986. IEEE Computer Society.
- [7] Lszl Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.
- [8] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702 – 732, 2004. Special Issue on FOCS 2002.
- [9] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 294–304, New York, NY, USA, 1993. ACM.
- [10] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, pcps, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27:804–915, June 1998.

- [11] Eric Blais. Improved bounds for testing juntas. In *Proceedings of the 11th international workshop, APPROX 2008, and 12th international workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques*, APPROX '08 / RANDOM '08, pages 317–330, Berlin, Heidelberg, 2008. Springer-Verlag.
- [12] Eric Blais. Testing juntas nearly optimally. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 151–158, New York, NY, USA, 2009. ACM.
- [13] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *IEEE Conference on Computational Complexity*, pages 210–220, 2011.
- [14] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *IEEE Conference on Computational Complexity*, pages 235–246, 2010.
- [15] Avrim Blum, Lisa Hellerstein, and Nick Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *J. Comput. Syst. Sci.*, 50:32–40, February 1995.
- [16] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47:549–595, December 1993.
- [17] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In *ICALP (1)*, pages 545–556, 2011.

- [18] Sourav Chakraborty, David Garca-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. *Electronic Colloquium on Computational Complexity (ECCC)*, pages 93–93, 2010.
- [19] Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Inf. Process. Lett.*, 90:301–305, June 2004.
- [20] Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *FOCS*, pages 549–558, 2007.
- [21] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM-APPROX*, pages 97–108, 1999.
- [22] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost np-complete. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:2–12, 1991.
- [23] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, pages 753–787, 2004.
- [24] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *STOC'02*, pages 474–483, 2002.
- [25] Oded Goldreich. On testing computability by small width obdds. In *APPROX-RANDOM*, pages 574–587, 2010.

- [26] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, July 1998.
- [27] Johan Håstad. Testing of the long code and hardness for clique. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, STOC '96*, pages 11–19, New York, NY, USA, 1996. ACM.
- [28] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.*, pages 545–557, 1992.
- [29] Kevin Matulef, Ryan O’Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing ± 1 -weight halfspace. In *APPROX-RANDOM*, pages 646–657, 2009.
- [30] Ilan Newman. Private vs. common random bits in communication complexity. 39:67–71, July 1991.
- [31] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM J. Disc. Math*, 16:2002, 2002.
- [32] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM J. Comput.*, 39(3):813–842, 2009.
- [33] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385 – 390, 1992.
- [34] Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5:73–205, February 2010.

- [35] Dana Ron and Gilad Tsur. On approximating the number of relevant variables in a function. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:78, 2011.
- [36] Ronitt Rubinfeld. Robust functional equations and their applications to program testing. *SIAM J. Comput.*, 28(6):1972–1997, 1999.
- [37] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
- [38] Luca Trevisan. Recycling queries in pcps and in linearity tests (extended abstract). In *STOC*, pages 299–308, 1998.
- [39] Gregory Valiant and Paul Valiant. Estimating the unseen: an $n/\log(n)$ -sample estimator for entropy and support size, shown optimal via new clts. In *STOC*, pages 685–694, 2011.
- [40] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.
- [41] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *Proc. 11th STOC*, pages 209–213, New York, NY, USA, 1979.