

# On Active and Passive Testing

Noga Alon\*

Rani Hod†

Amit Weinstein‡

December 1, 2013

## Abstract

Given a property of Boolean functions, what is the minimum number of queries required to determine with high probability if an input function satisfies this property? This is a fundamental question in Property Testing, where traditionally the testing algorithm is allowed to pick its queries among the entire set of inputs. Balcan et al. have recently suggested to restrict the tester to take its queries from a smaller, typically random, subset of the inputs. This model is called *active testing*, in resemblance of active learning. Active testing gets more difficult as the size of the set we can query from decreases, and the extreme case is when it is exactly the number of queries we perform (so the algorithm actually has no choice). This is known as *passive testing*, or testing from random examples.

In their paper, Balcan et al. have shown that active and passive testing of dictator functions is as hard as learning them, and requires  $\Theta(\log n)$  queries (unlike the classic model, in which it can be done in a constant number of queries). We extend this result to  $k$ -linear functions, proving that passive and active testing of them requires  $\Theta(k \log n)$  queries, assuming  $k$  is not too large.

Other classes of functions we consider are juntas, partially symmetric functions, linear functions, and low degree polynomials. For linear functions we provide tight bounds on the query complexity in both active and passive models (which asymptotically differ). The analysis for low degree polynomials is less complete and the exact query complexity is given only for passive testing. In both these cases, the query complexity for passive testing is essentially equivalent to that of learning.

For juntas and partially symmetric functions, that is, functions that depend on a small number of variables and potentially also on the Hamming weight of the input, we provide some lower and upper bounds for the different models. When the functions depend on a constant number of variables, our analysis for both families is asymptotically tight. Moreover, the family of partially symmetric functions is the first example for which the query complexities in all these models are asymptotically different.

Our methods combine algebraic, combinatorial, and probabilistic techniques, including the Talagrand concentration inequality and the Erdős–Rado results on  $\Delta$ -systems.

---

\*Sackler School of Mathematics and Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel and Institute for Advanced Study, Princeton, New Jersey, 08540. Email: [nogaa@tau.ac.il](mailto:nogaa@tau.ac.il). Research supported in part by an ERC Advanced grant, by a USA-Israeli BSF grant, by an ISF grant, by the Israeli I-Core program and by the Simonyi Fund.

†School of Mathematics, Georgia Tech, 686 Cherry st, Atlanta, GA 30332. E-mail: [rani.hod@math.gatech.edu](mailto:rani.hod@math.gatech.edu).

‡Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. Email: [amitw@tau.ac.il](mailto:amitw@tau.ac.il). Research supported in part by an ERC Advanced grant and by the Israeli Centers of Research Excellence (I-CORE) program.

# 1 Introduction

Property testing considers the following general problem: given a property  $\mathcal{P}$ , identify the minimum number of queries required to determine with high probability whether an input object has the property  $\mathcal{P}$  or whether it is “far” from  $\mathcal{P}$ . This question was first formalized by Rubinfeld and Sudan [23] in the context of Boolean functions.

**Definition 1** ([23]). Let  $\mathcal{P}$  be a family of Boolean functions and let  $\epsilon > 0$ . A  $q$ -query  $\epsilon$ -tester for  $\mathcal{P}$  is a randomized algorithm that queries an unknown function  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  on  $q$  inputs of its choice and

- (i) Accepts with probability at least  $2/3$  when  $f \in \mathcal{P}$ ;
- (ii) Rejects with probability at least  $2/3$  when  $f$  is  $\epsilon$ -far from  $\mathcal{P}$ , where  $f$  is  $\epsilon$ -far from  $\mathcal{P}$  if  $\text{dist}(f, g) := |\{x \in \mathbb{Z}_2^n \mid f(x) \neq g(x)\}| \geq \epsilon 2^n$  holds for every  $g \in \mathcal{P}$ .

We denote the minimal  $q$  such that a  $q$ -query  $\epsilon$ -tester for  $\mathcal{P}$  exists by  $Q_\epsilon(\mathcal{P})$ .

The main line of research in many works on property testing is to characterize  $Q_\epsilon(\mathcal{P})$  for various properties  $\mathcal{P}$ . An interesting distinction is identifying properties for which  $Q_\epsilon(\mathcal{P})$  is constant (i.e., independent of  $n$ ). For instance, linearity can be tested in a constant number of queries [13]; more generally, testing if a Boolean function is a polynomial of constant degree can be performed with a constant number of queries [1, 5, 7, 23]. Testing whether a function depends only on a constant number of its input variables (that is, if a function is a junta) can also be done with a constant number of queries [8, 9, 18].

In the definition above the algorithm can pick its  $q$  queries in the entire set  $\mathbb{Z}_2^n$ . Balcan et al. [3] suggested to restrict the tester to take its queries from a smaller, typically random, subset  $U \subseteq \mathbb{Z}_2^n$ . This model is called active testing, in resemblance of active learning (see, e.g., [15]). Active testing gets more difficult as the size of  $U$  decreases, and the extreme case is when  $U$  is a set of  $q$  random points (so the algorithm actually has no choice). This is known as passive testing, or testing from random examples<sup>1</sup>, and was studied in [19, 20]. Formally, the next definition from [3] extends Definition 1 to active and passive testers.

**Definition 2.** Let  $\mathcal{P}$  be a family of Boolean functions and let  $\epsilon > 0$ . A  $u$ -sample  $q$ -query  $\epsilon$ -tester for  $\mathcal{P}$  is a randomized algorithm that, given a subset  $U \subseteq \mathbb{Z}_2^n$  of size  $|U| = u$ , drawn uniformly at random, queries an unknown function  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  on  $q$  inputs from  $U$  and

- (i) Accepts with probability at least  $2/3$  when  $f \in \mathcal{P}$ ;
- (ii) Rejects with probability at least  $2/3$  when  $f$  is  $\epsilon$ -far from  $\mathcal{P}$ .

We denote by  $Q_\epsilon^a(\mathcal{P}, u)$  the minimal  $q$  such that a  $u$ -sample  $q$ -query  $\epsilon$ -tester for  $\mathcal{P}$  exists, and by  $Q_\epsilon^p(\mathcal{P})$  the minimal  $q$  such that a  $q$ -sample  $q$ -query  $\epsilon$ -tester (i.e., a passive  $\epsilon$ -tester) for  $\mathcal{P}$  exists.

We are usually interested in poly( $n$ )-sample testers; for simplicity, we omit the sample size  $u$  from our notation when this is the case.

The following inequality from [3] shows the relation between the query complexity of the different testing models.

---

<sup>1</sup>Although the examples could be drawn from a general probability distribution, in this work we focus on the uniform distribution.

**Proposition 3** ([3, Theorem A.4.]). *For every property  $\mathcal{P}$  and for every  $\epsilon > 0$ ,  $Q_\epsilon(\mathcal{P}) \leq Q_\epsilon^a(\mathcal{P}) \leq Q_\epsilon^p(\mathcal{P})$ .*

To provide a simple upper bound on the query complexity of passive testing, we refer to the more difficult problem of proper passive learning. The most common model of passive learning is PAC-learning, introduced by Valiant [26].

**Definition 4.** Let  $\mathcal{P}$  be a family of Boolean functions. A  $q$ -query  $\epsilon$ -learning algorithm for  $\mathcal{P}$  is a randomized algorithm that, given  $q$  random queries from an unknown function  $f \in \mathcal{P}$ , outputs a Boolean function  $g : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  such that  $g$  is  $\epsilon$ -close to  $f$  with probability at least  $2/3$  (the underlying probability space is the random queries and the coin tosses of the algorithm). The algorithm is called *proper* if it always returns some  $g \in \mathcal{P}$ . We denote the minimal  $q$  such that a proper  $q$ -query  $\epsilon$ -learning algorithm for  $\mathcal{P}$  exists by  $Q_\epsilon^\ell(\mathcal{P})$ .

The number of queries needed to properly learn a Boolean function essentially bounds from above the number of queries needed to test it; given the output of a proper learning algorithm, it remains to verify that the input function is indeed close to it. More formally, we have the following proposition.

**Proposition 5** ([19, Proposition 3.1.1]). *For every property  $\mathcal{P}$  and for every  $\epsilon > 0$ ,  $Q_\epsilon^p(\mathcal{P}) \leq Q_{\epsilon/2}^\ell(\mathcal{P}) + O(1/\epsilon)$ .*

This proposition is often used together with the following known upper bound.

**Fact 6.** *For every family of Boolean functions  $\mathcal{P}$ ,  $Q_\epsilon^\ell(\mathcal{P}) = O(\frac{1}{\epsilon} \log |\mathcal{P}|)$ .*

For the sake of simplicity, we focus on a constant  $\epsilon$  (say,  $\epsilon = 0.001$ ) throughout the rest of this paper. This allows us to drop the subscript  $\epsilon$  from our notation when possible (e.g., we write  $Q(\mathcal{P})$  instead of  $Q_\epsilon(\mathcal{P})$ ).

## 1.1 Our results

In [3] it was shown that active testing of dictator functions (i.e., functions that only depend on a single input variable) requires  $\Theta(\log n)$  queries. Our first result extends this to the family of  $k$ -linear functions; that is, the family of sums of  $k$  variables over  $\mathbb{Z}_2$ . Let  $\text{Lin}_k$  denote this family.

**Theorem 7.** *Active or passive testing of Boolean  $k$ -linear functions requires  $\Theta(k \log n)$  queries, for all  $k = O(\log n / \log \log n)$ .*

Theorem 7 and its proof imply a lower bound for active testing of superfamilies of  $k$ -linear functions, such as  $k$ -juntas and  $(n - k)$ -symmetric functions. A function is called  $k$ -junta if it depends on at most  $k$  of its input variables, referred to as the influential variables (e.g., a dictator function is a 1-junta). We denote the family of  $k$ -juntas by  $\text{Jun}_k$ . Partially symmetric functions are a generalization of juntas, where the remaining variables can influence the output of the function, but only in a symmetric manner.

**Definition 8** (Partially symmetric functions [12]). For a subset  $T \subseteq [n] := \{1, \dots, n\}$ , a function  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  is called  $T$ -symmetric if permuting the labels of the variables of  $T$  does not change the function. Moreover,  $f$  is called  $t$ -symmetric if there exists  $T \subseteq [n]$  of size at least  $t$  such that  $f$  is  $T$ -symmetric. We denote the family of  $t$ -symmetric functions by  $\text{Sym}_t$ .

Partially symmetric functions were introduced as part of the research of isomorphism testing [12], where it was shown that testing whether a function is  $(n - k)$ -symmetric for any  $k < n/10$  can be done using  $O(k \log k)$  queries. The special case of 2-symmetric functions has already been considered by Shannon in [24]. In addition to the  $\Omega(k \log n)$  lower bound for active testing of partially symmetric functions, we provide an upper bound as well as lower and upper bounds for passive testing (detailed in Table 1). In particular, we show that for a constant  $k$ , the family of partially symmetric functions demonstrates a significant gap among the three different testing scenarios and proper learning.

**Theorem 9.** *For a constant  $k$  we have*

$$\begin{aligned} Q(\text{Sym}_{n-k}) &= \Theta(1), \\ Q^a(\text{Sym}_{n-k}) &= \Theta(\log n), \\ Q^p(\text{Sym}_{n-k}) &= \Theta(n^{1/4} \log n), \\ Q^\ell(\text{Sym}_{n-k}) &= \Theta(\sqrt{n}). \end{aligned}$$

The last family of functions considered in this work is low degree polynomials, with special consideration given to linear functions. The following indicates that passive testing of degree  $d$  polynomials, denoted by  $\text{Pol}_d$ , is essentially as hard as properly learning them.

**Theorem 10.** *The query complexity of passive testing of degree  $d$  polynomials is  $\Theta(n^d)$ , for constant  $d$ .*

On the other hand, active testing can be done slightly more efficiently, at least for linear functions.

**Theorem 11.** *The query complexity of active testing of linear functions is  $\Theta(n/\log n)$ .*

Table 1 summarizes the results presented in this work for passive and active testing, as well as the best known query complexity for the classical model of property testing and proper learning.

Family	Classic ( $Q$ )	Active ( $Q^a$ )	Passive ( $Q^p$ )	Learning ( $Q^\ell$ )
Symmetric	$O(1)$	$O(1)$	$\Theta(n^{1/4})$	$\Theta(\sqrt{n})$
Linear	$O(1)$ [13]	$\Theta(n/\log n)$	$n + \Theta(1)$	$n + \Theta(1)$
$d$ -degree polynomials	$\Theta(2^d)$ [1, 7]		$\Theta(n^d)$	$\Theta(n^d)$
$k$ -linear	$O(k \log k)$ , $k - o(k)$ [9, 11]	$\Theta(k \log n)$	$\Theta(k \log n)$	$\Theta(k \log n)$
$k$ -juntas	$O(k \log k)$ , $\Omega(k)$ [9, 10, 14]	$\Omega(k \log n)$	$\Omega(2^{k/2} + k \log n)$	$\Theta(2^k + k \log n)$
$(n - k)$ -symmetric	$O(k \log k)$ , $\Omega(k)$ [12]	$O(2^k k \log n)$ , $\Omega(k \log n)$	$O(n^{1/4} \sqrt{2^k k \log n})$ , $\Omega(n^{1/4} \sqrt{2^k + k \log n})$	$\Theta(\sqrt{n} 2^k)$

Table 1: Summary of best bounds, for  $k < \log n / \log \log n$

The rest of the paper is organized as follows. The lower bound for active testing of  $k$ -linear functions, which applies to juntas and partially symmetric functions as well, is proved in Section 2 by establishing a general result for random subsets of abelian groups, proved by combining probabilistic

and combinatorial tools including the Talagrand inequality and the Erdős–Rado results on  $\Delta$ -systems. Section 3 provides the lower and upper bounds for active and passive testing of symmetric and partially symmetric functions, as described in Table 1. The results concerning low degree polynomials and linear functions in particular are presented in Section 4. Concluding remarks and open problems are in Section 5. The proofs in Sections 3 and 4 are also based on probabilistic, combinatorial, and algebraic techniques.

## 2 $k$ -linear functions

Theorem 7 states that the query complexity of active or passive testing of  $k$ -linear functions is  $\Theta(k \log n)$ . The upper bound can be obtained by applying Propositions 3 and 5, and Fact 6, given that there are exactly  $\binom{n}{k}$  different  $k$ -linear functions.

In order to prove a lower bound for active testing of  $k$ -linear functions, we use the following lemma, which is an adaptation of the tools used in [3] to prove active testing lower bounds (specifically, Theorem 6.6 and Lemma B.1 *ibid*).

**Lemma 12** ([3]). *Let  $\mathcal{P}$  be a nontrivial property<sup>2</sup> of Boolean functions and let  $\pi$  be a distribution supported on  $\mathcal{P}$ . Given a set  $S = \{x_1, x_2, \dots, x_q\}$  of  $q$  queries and a vector  $y \in \mathbb{Z}_2^q$ , define*

$$\pi_S(y) = \Pr_{f \sim \pi} [f(x_i) = y_i \text{ for } i = 1, 2, \dots, q].$$

*Choose at random a set  $U$  of  $u$  samples, and suppose that with probability at least  $\frac{3}{4}$ , every set  $S \subseteq U$  of  $q$  queries and every  $y \in \mathbb{Z}_2^q$  satisfy  $\pi_S(y) < \frac{6}{5}2^{-q}$ . Then,  $Q^a(\mathcal{P}, u) \geq q$ .*

The proof is based on the fact that, under the assumptions of the lemma,  $q$  queries do not suffice to distinguish between a function from the distribution  $\pi$  and a uniform random Boolean function.

According to Lemma 12, our goal is therefore to show that when we choose a random  $k$ -linear function, querying it at  $o(k \log n)$  queries chosen from a random space will appear rather random. To this end we use Lemma 18, which, roughly speaking, assures us that the probability of seeing a given output vector is very concentrated around the expectation. The proof of the lemma uses the Talagrand inequality (with an extra twist) and the Erdős–Rado  $\Delta$ -systems method. Lemma 18, its proof, and the tools used appear in Section 2.1.

The following theorem provides a lower bound for active testing of  $k$ -linear functions, completing the proof of Theorem 7 (assuming Lemma 18).

**Theorem 13.**  $Q^a(\text{Lin}_k, u) = \Omega(k \log n)$  as long as  $n \leq u \leq \exp(n^{1/7k})$ .

*Proof.* Define  $\pi$  to be the uniform distribution over the  $k$ -linear functions. In particular,  $\pi$  is the distribution obtained by choosing distinct  $i_1, i_2, \dots, i_k \in [n]$  uniformly at random and returning the function  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  defined by  $f(x) = x_{i_1} + x_{i_2} + \dots + x_{i_k}$ . Fix  $S$  to be a set of  $q$  vectors in  $\mathbb{Z}_2^n$ . This set can be viewed as a  $q \times n$  Boolean-valued matrix. We write  $c_1(S), \dots, c_n(S)$  to represent the columns of this matrix. For any  $y \in \mathbb{Z}_2^q$ ,

$$\pi_S(y) = \binom{n}{k}^{-1} \left| \left\{ I \in \binom{[n]}{k} : \sum_{i \in I} c_i(S) = y \right\} \right|.$$

---

<sup>2</sup>In this context, a property  $\mathcal{P}$  is nontrivial if a random Boolean function is 0.01-far from satisfying  $\mathcal{P}$  with probability  $1 - o(1)$ .

By Lemma 12, to prove that  $Q^a(\text{Lin}_k, u) = \Omega(k \log n)$ , it suffices to show that when  $U$  is a set of  $u$  vectors chosen uniformly and independently at random from  $\mathbb{Z}_2^n$  and, say,  $q = \left(1 - \frac{1}{k}\right) \log \binom{n}{k} + k$ , then with probability at least  $\frac{3}{4}$ , every set  $S \subset U$  of size  $|S| = q$  and every  $y \in \mathbb{Z}_2^q$  satisfy  $\pi_S(y) \leq \frac{6}{5} 2^{-q}$ . To this end, we would like to show that  $\pi_S(y)$  is highly concentrated around  $\mathbb{E}[\pi_S(y)] = 2^{-q}$ .

To apply Lemma 18, consider the group  $G = \mathbb{Z}_2^q$  and let  $N = |G| = 2^q = 2^k \binom{n}{k}^{1-1/k}$ . By monotonicity, we assume  $u = \lfloor \exp(n^{1/7k}) \rfloor \geq n$  and let  $\lambda = \lceil qn^{1/7k} \rceil \geq q \log u$ . Now, conditions (1a) and (1b) of the lemma hold, and thus the probability that there exists a vector  $y \in \mathbb{Z}_2^q$  such that more than  $\frac{6}{5} \binom{n}{k} 2^{-q}$   $k$ -sets of columns of  $S$  sum to  $y$  is at most  $5 \cdot 2^{-\lambda}$ . Furthermore, when  $U$  is defined as above, we can apply the union bound over all  $y \in G$  and over all subsets  $S \subseteq U$  of size  $|S| = q$  to obtain

$$\Pr \left[ \exists S, y : \pi_S(y) > \frac{6}{5} 2^{-q} \right] \leq \binom{u}{q} \cdot 2^q \cdot 5 \cdot 2^{-\lambda} \leq \frac{u^q}{q!} \cdot 2^q \cdot 5 \cdot 2^{-q \log u} = o(1),$$

establishing the theorem.  $\square$

The above theorem immediately implies a lower bound for active testing of both  $k$ -juntas and  $(n - k)$ -symmetric functions. It can also be applied to show lower bounds for other concise representation families, such as small DNF formulas, small decision trees, small Boolean formulas, and small Boolean circuits (see [16]).

**Corollary 14.**  $Q^a(\text{Jun}_k) = \Omega(k \log n)$  and  $Q^a(\text{Sym}_{n-k}) = \Omega(k \log n)$  for  $k = O(\log n / \log \log n)$ .

In Section 3 we continue the investigation of active and passive testing of partially symmetric functions. The following proposition summarizes what we know about passive testing of  $k$ -juntas.

**Proposition 15.**  $\Omega(2^{k/2} + k \log n) \leq Q^p(\text{Jun}_k) \leq O(2^k + k \log n)$ .

*Proof.* The upper bound is obtained by applying Proposition 5 and Fact 6, as the number of  $k$ -juntas is  $\binom{n}{k} 2^{2k}$ . The lower bound is a combination of two separate bounds:<sup>3</sup>  $\Omega(k \log n)$  by Corollary 14 and  $\Omega(2^{k/2})$  for verifying that the input function is indeed a junta, even when the set of the influencing variables is known in advance. Indeed, assume we are given the input function with a promise that it is either a junta over the first  $k$  variables or a random function. Distinguishing between these two cases requires a pair of inputs agreeing on the first  $k$  variables, which is unlikely among  $o(2^{k/2})$  queries.  $\square$

## 2.1 Proof of main lemma

Before we state the formal lemma, we introduce the two following combinatorial and probabilistic tools used in the proof.

### Erdős–Rado $\Delta$ -systems

**Definition.** Let  $a, b$  be positive integers. We say that a family of  $a$  sets, each of size  $b$ , forms a  $\Delta$ -system of size  $a$  if they have pairwise the same intersection.

Erdős and Rado proved that every large enough family of sets contains a large  $\Delta$ -system.

**Theorem 16** ([17, Theorem 3]). *Let  $\mathcal{F}$  be a family of sets, each of size  $b$ . Then  $\mathcal{F}$  contains  $\Delta$ -system of size  $a$  whenever  $|\mathcal{F}| \geq (a - 1)^{b+1} b!$ .*

<sup>3</sup>Although Corollary 14 only holds for  $k = O(\log n / \log \log n)$ , for larger values of  $k$  its contribution to the lower bound is negligible.

### Talagrand's concentration inequality

In its general form, Talagrand's inequality is an isoperimetric inequality for product probability spaces. We use the following formulation from [22] (see also [2, 25]), suitable for showing that a random variable in a product space is concentrated around its expectation under two conditions:

**Theorem 17** ([22, page 81]). *Let  $X \geq 0$  be a non-trivial random variable, determined by  $n$  independent trials  $T_1, \dots, T_n$ . If there exist  $c, r > 0$  such that*

- (i)  *$X$  is  $c$ -Lipschitz: changing the outcome of one trial can affect  $X$  by at most  $c$ , and*
- (ii)  *$X$  is  $r$ -certifiable: for any  $s$ , if  $X \geq s$  then there is a set of at most  $rs$  trials whose outcomes certify that  $X \geq s$ ,*

then for any  $0 \leq t \leq \mathbb{E}[X]$ ,

$$\Pr[|X - \mathbb{E}[X]| > t + 60\sqrt{\tau}] < 4 \exp(-t^2/8\tau),$$

where  $\tau = c^2 r \mathbb{E}[X]$ .

We now state the main lemma.

**Lemma 18.** *Let  $G$  be an abelian group of order  $N$ , and let  $n \in \mathbb{N}$ . Consider a random sequence  $X = (x_1, x_2, \dots, x_n)$ , where each  $x_i \in G$  is chosen uniformly and independently at random (with repetitions). Fix  $y \in G$  and  $k \in \mathbb{N}$ , and let  $Y = |\mathcal{Y}|$ , where  $\mathcal{Y} = \left\{ I \in \binom{[n]}{k} : \sum_{i \in I} x_i = y \right\}$ . Let  $\lambda \geq 2 \log N$  be a positive integer and assume that*

$$\binom{n}{k} \geq 800 \ln 2 \cdot k N \lambda^{2k+1}; \text{ and} \tag{1a}$$

$$\binom{n}{k-1} \leq \frac{\lambda N}{k 2^k}. \tag{1b}$$

Then,

$$\Pr\left[|Y - \mathbb{E}[Y]| > \frac{1}{5} \mathbb{E}[Y]\right] < 5 \cdot 2^{-\lambda}.$$

*Proof.* This is trivial for  $k = 1$ , so we assume  $k \geq 2$ . We would like to use Talagrand's inequality to prove that  $Y$  is concentrated around  $\mathbb{E}[Y]$ , but  $Y$  does not satisfy the Lipschitz condition necessary for its application. Let us thus define  $\hat{Y} = |\hat{\mathcal{Y}}|$ , where  $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$  is maximal such that, for all  $j \in [n]$ ,  $x_j$  belongs to at most  $c$  sets  $I \in \hat{\mathcal{Y}}$ ; the exact value of  $c$  will be determined later.

First we bound the probability that  $\hat{Y} \neq Y$ . Let  $\mathcal{Y}_j = \left\{ I \in \binom{[n] \setminus \{j\}}{k-1} : I \cup \{j\} \in \mathcal{Y} \right\}$ . By Theorem 16, there exists a  $\Delta$ -system  $\mathcal{Z}_j \subseteq \mathcal{Y}_j$  of size

$$|\mathcal{Z}_j| \geq 1 + \sqrt[k]{|\mathcal{Y}_j| / (k-1)!} > |\mathcal{Y}_j|^{1/k} e/k,$$

where every two distinct  $I_1, I_2 \in \mathcal{Z}_j$  have the same intersection  $K_j = I_1 \cap I_2$ . Thus,  $\mathcal{Z}'_j = \{I \setminus K_j : I \in \mathcal{Z}_j\}$  is a collection of  $s_j = |\mathcal{Z}'_j| = |\mathcal{Z}_j|$  disjoint  $k'$ -sets such that  $\sum_{i \in I} x_i = z$  for all  $I \in \mathcal{Z}'_j$ , where  $k' = k - 1 - |K_j| \leq k - 1$  and  $z = y - x_j - \sum_{i \in K_j} x_i$ .

Consider the event  $E_z(s)$ , defined as the existence of a collection of  $s$  disjoint  $k'$ -subsets of  $X$  that all sum to the same element  $z \in G$ . Then,

$$\Pr[E_z(s)] \leq \binom{n}{\underbrace{k', k', \dots, k'}_s} N^{-s} \leq \frac{1}{s!} \binom{n}{k'}^s N^{-s} \leq \left( \frac{e}{sN} \binom{n}{k-1} \right)^s \leq \left( \frac{e\lambda}{sk2^k} \right)^s,$$

and thus we have, for the choice of  $c = \lambda^k$ ,

$$\begin{aligned} \Pr[Y > \hat{Y}] &\leq \Pr[\exists j \in [n] : |\mathcal{Y}_j| > c] \leq \Pr[\exists j \in [n] : s_j > c^{1/k} e/k] \\ &\leq \Pr[\exists z \in G : E_z(e\lambda/k)] \leq N (2^{-k})^{e\lambda/k} = 2^{\log N - e\lambda} < 2^{-2\lambda}. \end{aligned} \quad (2)$$

This also serves to show that  $\mathbb{E}[Y]$  and  $\mathbb{E}[\hat{Y}]$  are very close, since

$$\mathbb{E}[Y - \hat{Y}] \leq \max(Y - \hat{Y}) \cdot \Pr[Y > \hat{Y}] \leq \binom{n}{k} 2^{-2\lambda} \leq \binom{n}{k-1} 2^{-2\lambda} \leq \left( \frac{\lambda N}{k2^k} \right)^2 2^{-2\lambda} \leq \frac{\lambda^2 2^{-\lambda}}{64} < \frac{1}{32}. \quad (3)$$

Next we apply Talagrand's inequality to bound the deviation of  $\hat{Y}$  from  $\mathbb{E}[\hat{Y}]$ . By definition,  $\hat{Y}$  is  $c$ -Lipschitz; moreover, to prove that  $\hat{Y} \geq s$  we only need to reveal  $s$   $k$ -sets, i.e., reveal  $x_i$  for at most  $ks$  values of  $i$ . For every choice of  $I \in \binom{[n]}{k}$ ,  $\sum_{i \in I} x_i$  is a random element of  $G$  and thus  $\mathbb{E}[Y] = \binom{n}{k}/N \geq 800 \ln 2 \cdot k\lambda^{2k+1}$ .

Set  $\tau = c^2 k \mathbb{E}[\hat{Y}]$ . By Theorem 17,

$$\begin{aligned} \Pr\left[|\hat{Y} - \mathbb{E}[\hat{Y}]| > \frac{1}{10} \mathbb{E}[Y] + 60\sqrt{\tau}\right] &\leq 4 \exp\left(-\mathbb{E}[Y]^2/800\tau\right) \\ &\leq 4 \exp\left(-\mathbb{E}[Y]/800c^2k\right) < 4 \exp\left(-\lambda^{2k+1} \ln 2/c^2\right) = 4 \cdot 2^{-\lambda}. \end{aligned} \quad (4)$$

Putting (2), (3) and (4) together,

$$\begin{aligned} \Pr\left[|Y - \mathbb{E}[Y]| > \frac{1}{5} \mathbb{E}[Y]\right] &\leq \Pr[Y > \hat{Y}] + \Pr\left[|\hat{Y} - \mathbb{E}[Y]| > \frac{1}{5} \mathbb{E}[Y]\right] \\ &\leq 2^{-\lambda} + \Pr\left[|\hat{Y} - \mathbb{E}[\hat{Y}]| > \frac{1}{5} \mathbb{E}[Y] - \frac{1}{32}\right] < 5 \cdot 2^{-\lambda}, \end{aligned}$$

under the condition  $\frac{1}{5} \mathbb{E}[Y] - \frac{1}{32} \geq \frac{1}{10} \mathbb{E}[Y] + 60\sqrt{\tau}$ , satisfied whenever  $\lambda \geq 650$ .  $\square$

### 3 Partially symmetric functions

The following proposition determines the number of queries needed for passive and active testing of symmetric Boolean functions. Although these results are a special case of partially symmetric functions, we feel that this serves as an introduction and provides some intuition.

**Proposition 19.**  $Q^p(\text{Sym}_n) = \Theta(n^{1/4})$  and  $Q^a(\text{Sym}_n) = O(1)$ .

*Proof.* A symmetric function is characterized by its layers of different Hamming weight. For each Hamming weight between 0 and  $n$ , the function outputs a consistent value. To test symmetry given a function, it suffices to randomly choose an input  $x \in \mathbb{Z}_2^n$  and a permutation of it, and see if the output is consistent over the two inputs. Since the Hamming weight of  $x$  is distributed  $\text{Bin}(n, 1/2)$ ,

two random inputs share the same Hamming weight with probability  $\Theta(1/\sqrt{n})$  and hence we would need  $\Omega(n^{1/4})$  random samples in order to have even a single such pair. Naturally, among  $O(n^{1/4})$  random samples we can find several such pairs with random weights and verify the symmetry with good probability.

In the context of active testing, given a sample space of, say,  $u = n$  vectors we can easily find many pairs of vectors with the same Hamming weight, thus testing whether the input function is indeed symmetric can be done using a constant number of queries.  $\square$

*Remark.* Consider the following slight modification of the algorithms above. Instead of rejecting the input function upon the first example of it not being symmetric, we estimate its distance by counting the number of such examples among all pairs. This enables us to distinguish between a function that is  $\epsilon/2$ -close to being symmetric and one that is  $\epsilon$ -far, while maintaining the same query complexity. Such an algorithm is called a tolerant tester.

Some families of Boolean functions, such as symmetric and partially symmetric functions, have many pairs of functions which are close to one another. In these cases, the upper bound of Fact 6, which relies only on the size of the family, is not tight. We remedy this by proving the following refined version.

**Definition 20.** Let  $\mathcal{P}$  be a family of Boolean functions and let  $\epsilon > 0$ . Denote by  $\mathcal{I}_\epsilon(\mathcal{P})$  a subfamily of  $\mathcal{P}$  of maximal size such that every two distinct  $f, g \in \mathcal{I}_\epsilon(\mathcal{P})$  are  $\epsilon$ -far.

**Proposition 21.** Let  $\mathcal{P}$  be a family of Boolean functions and let  $\epsilon > 0$ . Then

$$\lfloor \log |\mathcal{I}_{2\epsilon}(\mathcal{P})| \rfloor \leq Q_\epsilon^\ell(\mathcal{P}) \leq \left\lceil \frac{64}{\epsilon} \ln |\mathcal{I}_{\epsilon/2}(\mathcal{P})| \right\rceil.$$

*Proof.* A proper learning algorithm for  $\mathcal{P}$  is required to return a function from  $\mathcal{P}$  that is  $\epsilon$ -close to the input function. Since functions in  $\mathcal{I}_{2\epsilon}(\mathcal{P})$  are  $2\epsilon$ -far from one another, the algorithm has to return a different output for each of them. Any deterministic algorithm making  $q$  queries can only have  $2^q$  different outputs, so if it performs less than  $\lfloor \log |\mathcal{I}_{2\epsilon}(\mathcal{P})| \rfloor$  queries, it must be wrong with probability at least  $1/2$ . A randomized algorithm for this problem can be viewed as a distribution over deterministic algorithms (as the queries are chosen randomly and the algorithm is non-adaptive), and therefore cannot improve the success probability beyond  $1/2$ .

Next, consider the following learning algorithm: given an input function  $f \in \mathcal{P}$ , return the function  $g \in \mathcal{I}_{\epsilon/2}(\mathcal{P})$  that agrees with  $f$  on as many queries as possible out of  $q = \left\lceil (64/\epsilon) \ln |\mathcal{I}_{\epsilon/2}(\mathcal{P})| \right\rceil$  random queries. By definition,  $f$  is  $\epsilon/2$ -close to some  $f' \in \mathcal{I}_{\epsilon/2}(\mathcal{P})$ ; therefore,  $f$  and  $f'$  disagree on each query with probability at most  $\epsilon/2$ , independently. The total number of disagreements is thus dominated by a  $\text{Bin}(q, \epsilon/2)$  random variable and hence with high probability they disagree on fewer than  $3\epsilon q/4$  queries. Using a similar argument, a function  $h \in \mathcal{I}_{\epsilon/2}(\mathcal{P})$  that is  $\epsilon$ -far from  $f$  will disagree with  $f$  on more than  $3\epsilon q/4$  queries with probability at least  $1 - \exp(-\epsilon q/32) = 1 - |\mathcal{I}_{\epsilon/2}(\mathcal{P})|^{-2}$ . By the union bound, with high probability no such  $h$  will outperform  $f'$  and thus the algorithm will return a function that is  $\epsilon$ -close to  $f$  (the obvious candidate being  $f'$ ).  $\square$

**Corollary 22.**  $Q^\ell(\text{Sym}_{n-k}) = \Theta(\sqrt{n-k}2^k)$  and, in particular,  $Q^\ell(\text{Sym}_n) = \Theta(\sqrt{n})$ .

*Proof.* The binomial distribution is concentrated around its center, and in particular for every constant  $\delta > 0$ , the middle  $O(\sqrt{n-k})$  layers account for  $1 - \delta$  of the weight. Moreover, the weight ratio between any pair of these layers is bounded by a constant. Therefore, once we consider only the middle layers, two random functions are  $\delta$ -close with exponentially small probability (in  $2^k \sqrt{n-k}$ ). Hence we get  $|\mathcal{I}_\delta(\text{Sym}_{n-k})| = 2^{\Theta(2^k \sqrt{n-k})}$  and therefore, for our fixed  $\epsilon$ , the result follows from Proposition 21.  $\square$

The following proposition provides an upper bound for the query complexity of passive and active testing of partially symmetric functions.

**Proposition 23.**  $Q^p(\text{Sym}_{n-k}) = O\left(n^{1/4}2^{k/2}\sqrt{k\log n}\right)$  and  $Q^a(\text{Sym}_{n-k}) = O\left(2^k k \log n\right)$ , for  $k = o(\log n)$ .

*Proof.* We begin with a passive testing algorithm. Let  $f$  be the tested Boolean function. Our algorithm asks  $q = c \cdot n^{1/4}2^{k/2}\sqrt{k\log n}/\epsilon$  queries, and if the results obtained are consistent with  $f$  being  $(n-k)$ -symmetric it accepts, otherwise it rejects. It remains to show that if  $f$  is  $\epsilon$ -far from being  $(n-k)$ -symmetric the algorithm rejects with high probability. Assume this is the case and fix a  $k$ -set  $A \in \binom{[n]}{k}$  of variables. If we choose a random vector  $x$  and another random vector  $y$  obtained from  $x$  by permuting the elements in  $[n] \setminus A$  the probability that  $f(x) \neq f(y)$  is  $\Omega(\epsilon)$ . It is not too difficult to show that for an appropriate choice of  $c$ , with probability at least  $1 - n^{-k}/10$  our queries will contain more than  $c'k \log n/\epsilon$  pairs  $x, y$  which have the same Hamming weight and agree on  $A$ , and are random subject to these requirements. The detailed argument of this claim is omitted. The probability that none of these pairs will satisfy  $f(x) \neq f(y)$  is at most  $(1 - \Omega(\epsilon))^{c'k \log n/\epsilon} < n^{-k}/10$ . The union bound thus completes the argument.

The same argument implies that the query complexity of active testing is  $O\left(2^k k \log n\right)$  because the only information the passive algorithm above used are the results for the  $\Theta(q^2/\sqrt{n})$  pairs  $x, y$  which agree on their Hamming weight.  $\square$

The following proposition provides a lower bound for the query complexity of passive testing of partially symmetric functions. Note that it matches the upper bound, up to a constant factor, when  $k$  is constant.

**Proposition 24.**  $Q^p(\text{Sym}_{n-k}) = \Omega\left(n^{1/4}\left(2^{k/2} + \sqrt{k\log n}\right)\right)$ .

*Proof.* As in the proof of Proposition 15, we use a combination of two lower bounds. The first one,  $\Omega(n^{1/4}2^{k/2})$ , is required even when the identity of the  $k$  asymmetric variables is known in advance. Assuming we are given the promise that the input function is either  $(n-k)$ -symmetric and the asymmetric variables are the first  $k$  variables, or it is far from being partially symmetric, one still needs to verify the partial symmetry. The only way to verify it is by having pairs of inputs that share Hamming weight and agree on the values of the first  $k$  variables. However, we expect to see no such pairs if the number of queries is  $o\left(n^{1/4}2^{k/2}\right)$ .

The second part of the lower bound uses the  $\Omega(k \log n)$  bound of Theorem 13. We wish to show that distinguishing the sum of a random  $k$ -linear function and a random symmetric function cannot be distinguished from a random function, given  $q = o\left(n^{1/4}\sqrt{k\log n}\right)$  queries. Indeed, assume this many queries were performed and denote by  $H \subseteq \{0, 1, \dots, n\}$  the set of Hamming weights attained by at least two queries. A balls and bins argument shows that we expect only  $o(k \log n)$  queries whose Hamming weight lies in  $H$ . Due to the random symmetric function, the algorithm cannot extract any information from queries that have a unique Hamming weight. Say that we reveal to the algorithm the value of the random symmetric function on  $H$ . Now, the algorithm has  $o(k \log n)$  queries and it must distinguish between a  $k$ -linear function and a random function. Even if the algorithm were allowed to choose which queries to pick out of the initial set of  $q$  queries, the lower bound for active testing of  $k$ -linear functions indicates this cannot be done.  $\square$

Theorem 9 follows from Propositions 23 and 24 and Corollaries 14 and 22, as well as the results of [12].

## 4 Low degree polynomials

We prove Theorem 10 for a more general case, allowing  $1 \leq d \leq n^{1/3}$ . Let  $\binom{n}{\leq d} = \sum_{i=0}^d \binom{n}{i}$  be the number of monomials of degree at most  $d$ . Note that for constant  $d$ , we have  $\binom{n}{\leq d} = \Theta(n^d)$ .

**Theorem** (Restatement of Theorem 10).  $Q^p(\text{Pol}_d) = \Theta(\binom{n}{\leq d})$ .

*Proof.* The number of polynomials of degree  $d$  is  $2^{\binom{n}{\leq d}}$ , hence by Fact 6 and Proposition 5, passive testing can be done using  $O(\binom{n}{\leq d})$  queries. We now show a lower bound of  $\Omega(\binom{n}{\leq d})$  queries.

Let  $x_1, x_2, \dots, x_q \in \mathbb{Z}_2^n$  be the set of  $q = \lfloor \binom{n}{\leq d} / 2e \rfloor$  random queries performed by a passive tester. For  $i = 1, \dots, q$ , define  $y_i \in \mathbb{Z}_2^{\binom{n}{\leq d}}$  to be the  $d$ -evaluation of  $x_i$ , that is, the evaluations of all possible monomials of degree at most  $d$  at  $x_i$ . It suffices to show that  $\{y_i\}_{i=1}^q$  are most likely linearly independent to conclude that any testing algorithm performs badly; indeed, since the  $\binom{n}{\leq d}$  monomials serve as a basis to  $\text{Pol}_d$ ,  $\{y_i\}_{i=1}^q$  being linearly independent implies that every possible output  $(f(x_1), \dots, f(x_q)) \in \mathbb{Z}_2^q$  is equally likely when choosing a random  $f \in \text{Pol}_d$ , so the tester sees a uniform distribution and therefore cannot decide.

In order to show that, with high probability, these vectors are linearly independent, we bound the probability that  $y_i$  is spanned by  $y_1, \dots, y_{i-1}$ , and then apply the union bound to show that none of these events is likely to occur. Let  $V_i = \text{span}\{y_1, \dots, y_{i-1}\}$  be the linear space spanned by the first  $i-1$  vectors. By Lemma 4 from [6], since

$$\dim V_i \leq i-1 < q \leq \binom{n}{\leq d} / 2e \leq \sum_{i=0}^d \binom{n(1-1/d)}{i},$$

no more than  $2^{n(1-1/d)}$   $d$ -evaluations of vectors from  $\mathbb{Z}_2^n$  reside in  $V_i$ . Thus,  $\Pr[y_i \in V_i] \leq 2^{-n/d}$  and, by the union bound,  $\Pr[\exists i. y_i \in V_i] \leq q \cdot 2^{-n/d} = o(1)$  for  $d \leq n^{1/3}$ .  $\square$

We now focus on linear functions, for which we determine the passive query complexity up to an additive constant term. We slightly abuse notation by using  $\text{Pol}_1$  to denote the family of linear functions, even though degree 1 polynomials include both linear and affine functions.

**Proposition 25.**  $Q^p(\text{Pol}_1) = n + \Theta(1)$ .

*Proof.* As in the proof of Theorem 10, a linearly independent query set is useless for the testing algorithm. Let  $x_1, x_2, \dots, x_q$  be a sequence of  $q \leq n$  queries and define  $X_i$  to be the event that  $x_i \in \text{span}\{x_1, \dots, x_{i-1}\}$ . The probability that some linear dependency exists among the  $q$  queries can be bounded by

$$2^{q-1-n} = \Pr[X_q] \leq \Pr\left[\bigcup_{i=1}^q X_i\right] \leq \sum_{i=1}^q \Pr[X_i] = \sum_{i=1}^q 2^{i-1-n} < 2^{q-n}.$$

For  $q > n$ , surely any set of  $q$  queries is linearly dependent.

Given the computation above, a set of  $q \leq n-2$  queries is expected to be linearly dependent with probability smaller than  $1/4$ . On the other hand,  $n + O(1)$  queries are very likely to provide a basis for  $\mathbb{Z}_2^n$  and  $O(1)$  linear dependencies, so we can learn the unique linear function consistent with the basis and then verify it; if the function is  $\epsilon$ -far from linear, each additional query is inconsistent with the learned function with a constant probability.  $\square$

Active testing allows us to reduce the query complexity by a logarithmic factor, in comparison to passive testing. We first prove the following lemma, which is an extension of the analysis of the BLR test provided by Bellare et al. [4].

**Lemma 26.** *Given a function  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  that is  $\epsilon$ -far from being linear,*

$$\Pr_{x_1, x_2, \dots, x_{2k} \in \mathbb{Z}_2^n} [f(x_1) + \dots + f(x_{2k}) = f(x_1 + \dots + x_{2k})] \leq \frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^{2k-1}$$

*Proof.* Since  $f$  is  $\epsilon$ -far from being linear, all of its Fourier coefficients  $\{\hat{f}(S) : S \subseteq [n]\}$  are bounded from above by  $1 - 2\epsilon$ . Similar to the analysis for the case  $k = 1$ , the success probability of this test is:

$$\frac{1}{2} + \frac{1}{2} \sum_{S \subseteq [n]} \hat{f}(S)^{2k+1} \leq \frac{1}{2} + \frac{1}{2} \left( \max_{S \subseteq [n]} \hat{f}(S)^{2k-1} \right) \sum_{S \subseteq [n]} \hat{f}(S)^2 = \frac{1}{2} + \frac{1}{2} \max_{S \subseteq [n]} \hat{f}(S)^{2k-1} \leq \frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^{2k-1},$$

where the middle equality holds by Parseval's theorem.  $\square$

Unlike the BLR test, which uses the case  $k = 1$ , in the context of active testing we need  $k$  to be almost linear in  $n$ , hence little amplification is necessary.

**Theorem** (Restatement of Theorem 11).  $Q^a(\text{Pol}_1, u) = \Theta(n/\log u)$ , for  $u \geq n^2$ .

*Proof.* Similar to the previous theorem, we bound the number of queries from below by showing that one is not expected to find a linear dependency of size smaller than  $n/2 \log u$  among a set of  $u$  samples. The expected number of linear dependencies of size  $q$  is at most

$$\binom{u}{q} 2^{q-n} \leq \left( \frac{2e \cdot u}{q} \right)^q 2^{-n} \leq u^q 2^{-n} = 2^{q \log u - n} \leq 2^{-n/2},$$

assuming  $2e \leq q \leq n/2 \log u$ . By Markov's inequality, the probability of having such a linear dependency is  $o(1)$  and therefore  $\Omega(n/2 \log u)$  queries are needed.

Given an input function that is  $\epsilon$ -far from being linear, we use the test of Lemma 26 to identify this. Fix  $q = \lceil 3n/\log u \rceil$  and consider a specific query  $x$  from our sample set. Using a similar calculation as above, we expect

$$\binom{u}{q} 2^{-n} \geq u^q q^{-q} 2^{-n} \geq 2^{3n - n - q \log q} \geq 2^{n/2}$$

sets of  $q$  vectors to sum to  $x$ . By Lemma 26 the probability that  $f$  passes a single such test is at most  $\frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^{q-1} < \frac{1}{2} + \frac{1}{2}(1 - 2\epsilon)^{3n/\log u}$ . Since  $\epsilon$  is constant, for large enough  $n$  this is smaller than 0.9, thus repeating the test a constant number of times reduces the probability of  $f$  passing all of them to less than 1/3 (obviously we never reject a linear function).  $\square$

## 5 Discussion

Throughout this work we have demonstrated new bounds for the number of queries needed for active and passive testing of several properties. In particular, we now know the amount of queries needed for testing  $k$ -linear functions in these new models.

A practical aspect of property testing algorithms that we did not cover is the actual running time, rather than just the number of queries performed, which was the only concern in this work.

Some of the algorithms we presented, especially those based on proper learning, have an exponential run-time complexity and it would be interesting to see whether active or passive testing can be done while maintaining polynomial running time.

Quite a few of the passive testing algorithms we provided can in fact be made tolerant; that is, they can be modified to accept functions close to satisfying the property while rejecting functions far from satisfying it (with some gap in between). For simplicity we did not explicitly show that. Such modifications usually do not have an effect on the asymptotic query complexity.

While Section 4 provides a tight analysis of active and passive testing of linear functions, for low degree polynomials our analysis is only tight for passive testing. Extrapolating based on the behavior of linear functions, it seems natural to expect that the query complexity of active testing of low degree polynomials is asymptotically lower than passive testing, perhaps by a polylogarithmic factor. This question remains open at the moment.

Finally we mention that Lemma 18, used in the proof of Theorem 13, can be used in the study of a seemingly unrelated problem of exhibiting a very sharp cutoff phenomenon in the mixing time of random walks in random (dense) Cayley graphs of abelian groups. Indeed, the lemma implies that for any abelian group  $G$  of order  $N$ , and for  $(\log N)^{1/3} \leq k \leq (\log N)^{1/2-\delta}$ , if we choose  $d \approx N^{1/(k-1)}$  random elements of  $G$ , then a random walk of length  $k-1$  in the resulting Cayley graph of  $G$  is far from being mixing (simply because we cannot reach most of the elements at all) while a random walk of length  $k$  is already mixing. While it is more interesting to study this problem for much sparser random Cayley graphs (see [21] for some related results), even the above statement for the dense case is interesting.

## References

- [1] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron, Testing low-degree polynomials over  $\text{GF}(2)$ . In: *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-APPROX '03)*, pp. 188–199, 2003.
- [2] Noga Alon and Joel H. Spencer, **The Probabilistic Method** (3rd Edition). Wiley, New York, 2008.
- [3] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang, Active property testing. In: *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*, pp. 21–30, 2012.
- [4] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos Kiwi, and Madhu Sudan, Linearity testing in characteristic two. *IEEE Transactions on Information Theory* **42**(6): 1781–1796, 1996.
- [5] László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity* **1**(1):3–40, 1991
- [6] Ido Ben-Eliezer, Rani Hod, and Shachar Lovett, Random low-degree polynomials are hard to approximate. *Computational Complexity* **21**(1): 63–81, 2012.
- [7] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman, Optimal testing of Reed-Muller codes. In: *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '10)*, pp. 488–497, 2010.

- [8] Eric Blais, Improved bounds for testing juntas. In: *Proceedings of the 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 12th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-APPROX '08)*, pp. 317–330, 2008.
- [9] Eric Blais, Testing juntas nearly optimally. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC' 09)*, pp. 151–158, 2009.
- [10] Eric Blais, Joshua Brody, and Kevin Matulef, Property Testing Lower Bounds via Communication Complexity. *Computational Complexity* **21**(2):311–358, 2012.
- [11] Eric Blais and Daniel Kane, Tight bounds for testing  $k$ -linearity. In: *Proceedings of the 15th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 16th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-APPROX '12)*, pp. 435–446, 2012.
- [12] Eric Blais, Amit Weinstein, and Yuichi Yoshida, Partially symmetric functions are efficiently isomorphism-testable. In: *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*, pp. 551–560, 2012.
- [13] Manuel Blum, Michael Luby, and Ronitt Rubinfeld, Self-testing/correcting with applications to numerical problems. In: *J. of Computer and System Sciences* **47**:549–595, 1993.
- [14] Hana Chockler and Dan Gutfreund, A lower bound for testing juntas. *Information Processing Letters* **90**(6):301–305, 2004.
- [15] David Cohn, Les Atlas, and Richard Ladner, Improving generalization with active learning. In: *Proceedings of the 15th International Conference on Machine Learning (ICML '94)*, pp. 201–221, 1994.
- [16] Ilias Diakonikolas, Homin Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco Servedio, and Andrew Wan. Testing for concise representations. In: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, pp. 549–558, 2007.
- [17] Paul Erdős and Richard Rado, Intersection theorems for systems of sets. *J. London Math. Soc.* **35**(1): 85–90, 1960.
- [18] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky, Testing juntas. *J. of Computer and System Sciences* **68**(4):753–787, 2004.
- [19] Oded Goldreich, Shafi Goldwasser, and Dana Ron, Property testing and its connection to learning and approximation. *J. of the ACM* **45**(4):653–750, 1998.
- [20] Michael Kearns and Dana Ron, Testing problems with sublearning sample complexity. *J. of Computer and System Sciences* **61**(3):428–456, 2000.
- [21] Eyal Lubetzky and Allan Sly, Cutoff phenomena for random walks on random regular graphs. *Duke Mathematical Journal* **153**(3):475–510, 2010.
- [22] Michael Molloy and Bruce Reed, **Graph colouring and the probabilistic method**, Springer, 2002.
- [23] Ronitt Rubinfeld and Madhu Sudan, Robust characterizations of polynomials with applications to program testing. *SIAM J. on Computing* **25**(2):252–271, 1996.

- [24] Claude E. Shannon, The synthesis of two-terminal switching circuits. *Bell System Technical Journal* **28**(1):59–98, 1949.
- [25] Michel Talagrand, Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l’IHÉS* **81**:73–203, 1995.
- [26] Leslie G. Valiant, A theory of the learnable. *Communications of the ACM* **27**(11):1134–1142, 1984.