Combinatorics, Probability and Computing (2016) 25, 1–20. © Cambridge University Press 2015
doi:10.1017/S0963548315000292

On Active and Passive Testing

NOGA ALON[1]†, RANI HOD[2] and AMIT WEINSTEIN[3]‡

[1]Sackler School of Mathematics and Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
and
School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540, USA
(e-mail: nogaa@tau.ac.il)

[2]School of Mathematics, Georgia Tech, 686 Cherry St., Atlanta, GA 30332, USA
(e-mail: rani.hod@math.gatech.edu)

[3]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
(e-mail: amitw@tau.ac.il)

Received 2 November 2013; revised 25 October 2015

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 or is 'far' from satisfying it? 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, Blais, Blum and Yang have recently suggested restricting the tester to take its queries from a smaller random subset of polynomial size of the inputs. This model is called *active testing*, and in the extreme case when the size of the set we can query from is exactly the number of queries performed, it is known as *passive testing*.

We prove that passive or active testing of k-linear functions (that is, sums of k variables among n over $\mathbb{Z}_2$) requires $\Theta(k \log n)$ queries, assuming k is not too large. This extends the case $k = 1$, (that is, dictator functions), analysed by Balcan, Blais, Blum and Yang.

We also consider other classes of functions including low-degree polynomials, juntas, and partially symmetric functions. Our methods combine algebraic, combinatorial, and probabilistic techniques, including the Talagrand concentration inequality and the Erdős–Rado theorem on Δ-systems.

2010 *Mathematics subject classification*: Primary 68Q17
Secondary 68R05

† Research supported in part by BSF grant 2012/107, by ISF grant 620/13, by the Israeli I-Core programme and by the Fund for Mathematics.
‡ Research supported in part by the Israeli I-Core programme.

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 [24] in the context of Boolean functions.

Definition 1.1 ([24]). Let $\mathcal{P}$ be a family of Boolean functions and let $\epsilon > 0$. A *q-query ϵ-tester* for $\mathcal{P}$ is a randomized algorithm that queries an unknown function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ 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 ϵ-far from $\mathcal{P}$, where f is ϵ-far from $\mathcal{P}$ if

$$\operatorname{dist}(f, g) := |\{x \in \mathbb{Z}_2^n \mid f(x) \neq g(x)\}| \geqslant \epsilon 2^n$$

holds for every $g \in \mathcal{P}$.

We denote the minimal q such that a q-query ϵ-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 whether a Boolean function is a polynomial of constant degree can be performed with a constant number of queries [1, 3, 7, 24]. 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, 19].

In the definition above, the algorithm can pick its q queries in the entire set $\mathbb{Z}_2^n$. Balcan, Blais, Blum and Yang [4] suggested restricting 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 analogy to active learning (see *e.g.* [16]). 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,[1] and was studied in [20, 21]. Formally, the next definition from [4] extends Definition 1.1 to active and passive testers.

Definition 1.2. Let $\mathcal{P}$ be a family of Boolean functions and let $\epsilon > 0$. A *u-sample q-query ϵ-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 $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ 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 ϵ-far from $\mathcal{P}$.

[1] Although the examples could be drawn from a general probability distribution, in this work we focus on the uniform distribution.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

The set U may be chosen with or without repetitions. For our purposes these two options will be equivalent, since in the parameters considered here the probability of a repetition is negligible.

We let $Q^a_\epsilon(\mathcal{P}, u)$ denote the minimal q such that a u-sample q-query ϵ-tester for $\mathcal{P}$ exists (∞ if u queries do not suffice), and by $Q^p_\epsilon(\mathcal{P})$ the minimal q such that a q-sample q-query ϵ-tester (*i.e.*, a passive ϵ-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 [4] shows the relation between the query complexity of the different testing models.

Proposition 1.3 ([4, Theorem A.4]). *For every property $\mathcal{P}$ and for every $\epsilon > 0$,*

$$Q_\epsilon(\mathcal{P}) \leqslant Q^a_\epsilon(\mathcal{P}) \leqslant Q^p_\epsilon(\mathcal{P}).$$

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 [27].

Definition 1.4. Let $\mathcal{P}$ be a family of Boolean functions. A *q-query ϵ-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 $g : \mathbb{Z}_2^n \to \mathbb{Z}_2$ such that g is ϵ-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 ϵ-learning algorithm for $\mathcal{P}$ exists by $Q^\ell_\epsilon(\mathcal{P})$.

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 1.5 ([20, Proposition 3.1.1]). *For every property $\mathcal{P}$ and for every $\epsilon > 0$,*

$$Q^p_\epsilon(\mathcal{P}) \leqslant Q^\ell_{\epsilon/2}(\mathcal{P}) + O(1/\epsilon).$$

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

Fact 1.6. *For every family of Boolean functions $\mathcal{P}$,*

$$Q^\ell_\epsilon(\mathcal{P}) = O\left(\tfrac{1}{\epsilon} \log |\mathcal{P}|\right). \qquad \square$$

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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

1.1. Our results

In [4] 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 Lin_k denote this family.

Theorem 1.7. *Active or passive testing of Boolean k-linear functions needs $\Theta(k \log n)$ queries, for all*

$$k \leqslant \frac{\log n}{10 \log \log n}.$$

Theorem 1.7 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 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 1.8 (Partially symmetric functions [12]). For $T \subseteq [n] := \{1, \ldots, n\}$, a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ 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 Sym_t.

Partially symmetric functions were introduced as part of the research of isomorphism testing [12, 14], 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 [25]. 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 1.9. *For a constant k we have*

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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Table 1. Summary of best bounds, for fixed ϵ and $k < \log n/(10 \log\log n)$

Family	Classic (Q)	Active (Q^a)	Passive (Q^p)	Learning (Q^ℓ)
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, 15]	$\Omega(k \log n)$	$\Omega(2^{k/2} + k \log n)$	$\Theta(2^k + k \log n)$
$(n-k)$-symmetric	$O(k \log k)$, $\Omega(k)$ [12]	$O(2^k k \log n)$, $\Omega(k \log n)$	$O(n^{1/4}\sqrt{2^k k \log n})$, $\Omega(n^{1/4}\sqrt{2^k + k \log n})$	$\Theta(\sqrt{n}2^k)$

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 Pol_d, is essentially as hard as properly learning them.

Theorem 1.10. *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 1.11. *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.

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 Δ-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.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

2. k-linear functions

Theorem 1.7 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 1.3 and 1.5, and Fact 1.6, 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 [4] to prove active testing lower bounds (specifically, Theorem 6.6 and Lemma B.1 in [4]).

Definition 2.1. A property $\mathcal{P}$ is called *ϵ-non-trivial* if a random Boolean function is ϵ-close to $\mathcal{P}$ with probability at most 0.01.

Lemma 2.2 ([4]). *Let $\mathcal{P}$ be an ϵ-non-trivial property of Boolean functions and let π be a distribution supported on $\mathcal{P}$. Given a set $S = \{x_1, x_2, \ldots, x_q\}$ of q queries and a vector $y \in \mathbb{Z}_2^q$, define*

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

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, $Q_\epsilon^a(\mathcal{P}, u) \geqslant q$. □

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 π and a uniform random Boolean function.

According to Lemma 2.2, 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 2.9, 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 Δ-systems method. Lemma 2.9, 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 1.7 (assuming Lemma 2.9).

Theorem 2.3. $Q^a(\mathrm{Lin}_k, u) = \Omega(k \log n)$ *for* $k \leqslant 0.1 \log n / \log \log n$, *when* $n \leqslant u \leqslant 2^{n^{1/7k}}$.

Proof. Define π to be the uniform distribution over the k-linear functions. In particular, π is the distribution obtained by choosing distinct $i_1, i_2, \ldots, i_k \in [n]$ uniformly at random and returning the function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ defined by $f(x) = x_{i_1} + x_{i_2} + \cdots + x_{i_k}$. 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 $c_1(S), \ldots, c_n(S)$ to represent the columns of this matrix. For any $y \in \mathbb{Z}_2^q$,

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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Since Lin_k is, say, 0.4-non-trivial, by Lemma 2.2, to prove that $Q^a(\mathrm{Lin}_k, u) = \Omega(k \log n)$, 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, $q = (1 - \frac{1}{k}) \log \binom{n}{k} + k$, 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 $\pi_S(y) \leqslant \frac{6}{5} 2^{-q}$. 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 2.9 (below), 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 $u = \lfloor 2^{n^{1/7k}} \rfloor \geqslant n$ and let $\lambda = \lceil q n^{1/7k} \rceil \geqslant q \log u$. Now, for sufficiently large n, conditions (2.3a) and (2.3b) of the lemma hold. Indeed, to prove the first inequality note that

$$800 \ln 2 \cdot kN\lambda^{2k+1} = 800 \ln 2 \binom{n}{k}^{1-1/k} k 2^k \lceil q n^{1/7k} \rceil^{2k+1} \leqslant 800 \binom{n}{k}^{1-1/k} \cdot (2qn^{1/7k})^{2k+1}. \quad (2.1)$$

Since $k < 0.1 \log n / \log \log n$ and $q < 2k \log n < (\log n)^2$, we have

$$800(2qn^{1/7k})^{2k+1} = o(\sqrt{n}) < n/k < \binom{n}{k}^{1/k}.$$

Therefore, the right-hand-side of (2.1) is smaller than $\binom{n}{k}$, establishing (2.3a).

To prove the second inequality note that

$$\frac{\lambda N}{k 2^k} = \frac{\lambda \binom{n}{k}^{1-1/k}}{k} \geqslant \frac{q n^{1/7k} \binom{n}{k}}{\binom{n}{k}^{1/k} k} = \frac{(n-k+1) q n^{1/7k}}{k \binom{n}{k}^{1/k} k} \binom{n}{k-1}. \quad (2.2)$$

However,

$$\frac{(n-k+1) q n^{1/7k}}{k \binom{n}{k}^{1/k} k} \geqslant \Omega\left(\frac{nk \log n}{nk} \right) > 1,$$

and therefore the right-hand-side of (2.2) is bigger than $\binom{n}{k-1}$, proving (2.3b).

Thus, for any fixed vector $y \in \mathbb{Z}_2^q$, the probability 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

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

establishing the theorem. □

The above theorem and its proof immediately imply a lower bound for active testing of both k-juntas and $(n-k)$-symmetric functions. This 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 [17]).

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Corollary 2.4.

$$Q^a(\mathrm{Jun}_k) = \Omega(k \log n) \quad \textit{and} \quad Q^a(\mathrm{Sym}_{n-k}) = \Omega(k \log n)$$

for $k = O(\log n / \log\log n)$.

Proof. The same distribution π from the proof of Theorem 2.3 (uniform distribution over the k-linear functions) is supported on k-juntas (resp. $(n-k)$-symmetric functions), and these properties, too, are still 0.4-non-trivial. □

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 2.5. $\Omega(2^{k/2} + k \log n) \leqslant Q^p(\mathrm{Jun}_k) \leqslant O(2^k + k \log n)$. □

Proof. The upper bound is obtained by applying Proposition 1.5 and Fact 1.6, as the number of k-juntas is $\binom{n}{k}2^{2^k}$. The lower bound is a combination of two separate bounds:[2] $\Omega(k \log n)$ by Corollary 2.4 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 random junta over the first k variables or a random function. Distinguishing between these two cases is impossible unless we have a pair of inputs agreeing on the first k variables; among less than $\frac{1}{2}2^{k/2}$ queries, we get such a pair with probability at most

$$\frac{1}{2}\left(\frac{1}{2}2^{k/2}\right)^2 \cdot 2^{-k} = 1/8.$$ □

2.1. Proof of the main lemma

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

2.1.1. Erdős–Rado Δ-systems.

Definition 2.6. Let a, b be positive integers. We say that a family of a sets, each of size b, forms a Δ-*system* of size a if all pairs have the same intersection.

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

Theorem 2.7 ([18, Theorem 3]). *Let $\mathcal{F}$ be a family of sets, each of size b. Then $\mathcal{F}$ contains Δ-system of size a whenever $|\mathcal{F}| \geqslant (a-1)^{b+1}b!$.* □

[2] Although Corollary 2.4 only holds for $k = O(\log n / \log\log n)$, for larger values of k its contribution to the lower bound is negligible.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

2.1.2. 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 [23] (see also [2, 26]), suitable for showing that a random variable in a product space is concentrated around its expectation under two conditions.

Theorem 2.8 ([23, p. 81]). *Let $X \geqslant 0$ be a non-trivial random variable, determined by n independent trials $T_1, \ldots, T_n$. 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 $X \geqslant s$ then there is a set of at most rs trials whose outcomes certify that $X \geqslant s$,*

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

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

where $\tau = c^2 r \mathbb{E}[X]$. □

We now state the main lemma.

Lemma 2.9. *Let G be an abelian group of order N, and let $n \in \mathbb{N}$. Consider a random sequence $X = (x_1, x_2, \ldots, x_n)$, 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 $\lambda \geqslant 2 \log N$ be a positive integer and assume that

$$\binom{n}{k} \geqslant 800 \ln 2 \cdot kN\lambda^{2k+1}, \quad \text{and} \tag{2.3a}$$

$$\binom{n}{k-1} \leqslant \frac{\lambda N}{k2^k}. \tag{2.3b}$$

Then,

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

Proof. For $k = 1$ we have $Y \sim \mathrm{Bin}(n, 1/N)$ and the result is implied by Chernoff's inequality, so we henceforth assume $k \geqslant 2$. 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 therefore 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

$$\mathcal{Y}_j = \left\{ I \in \binom{[n] \setminus \{j\}}{k-1} : I \cup \{j\} \in \mathcal{Y} \right\}.$$

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

By Theorem 2.7, there exists a Δ-system $\mathcal{Z}_j \subseteq \mathcal{Y}_j$ of size

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

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 $k' = k - 1 - |K_j| \leqslant k-1$ 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,

$$\mathbb{P}[E_z(s)] \leqslant \underbrace{\binom{n}{k', k', \ldots, k'}}_{s} N^{-s} \leqslant \frac{1}{s!}\binom{n}{k'}^s N^{-s} \leqslant \left(\frac{e}{sN}\binom{n}{k-1}\right)^s \leqslant \left(\frac{e\lambda}{sk2^k}\right)^s,$$

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

$$\begin{aligned}\mathbb{P}[Y > \hat{Y}] &\leqslant \mathbb{P}[\exists j \in [n] : |\mathcal{Y}_j| > c] \leqslant \mathbb{P}[\exists j \in [n] : s_j > c^{1/k} e/k] \\ &\leqslant \mathbb{P}[\exists z \in G : E_z(e\lambda/k)] \leqslant N(2^{-k})^{e\lambda/k} = 2^{\log N - e\lambda} < 2^{-2\lambda}. \end{aligned} \tag{2.4}$$

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

$$\begin{aligned}\mathbb{E}[Y - \hat{Y}] &\leqslant \max(Y - \hat{Y}) \cdot \mathbb{P}[Y > \hat{Y}] \\ &\leqslant \binom{n}{k} 2^{-2\lambda} \leqslant \binom{n}{k-1}^2 2^{-2\lambda} \\ &\leqslant \left(\frac{\lambda N}{k2^k}\right)^2 2^{-2\lambda} \leqslant \frac{\lambda^2 2^{-\lambda}}{64} < \frac{1}{32}. \end{aligned} \tag{2.5}$$

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 $\hat{Y} \geqslant s$ we only need to reveal s k-sets, that is, 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

$$\mathbb{E}[Y] = \binom{n}{k}/N \geqslant 800 \ln 2 \cdot k\lambda^{2k+1}.$$

Set $\tau = c^2 k \mathbb{E}[\hat{Y}]$. By Theorem 2.8,

$$\begin{aligned}\mathbb{P}\left[|\hat{Y} - \mathbb{E}[\hat{Y}]| > \frac{1}{10}\mathbb{E}[Y] + 60\sqrt{\tau}\right] &\leqslant 4\exp(-\mathbb{E}[Y]^2/800\tau) \\ &\leqslant 4\exp(-\mathbb{E}[Y]/800c^2k) < 4\exp(-\lambda^{2k+1}\ln 2/c^2) \\ &= 4 \cdot 2^{-\lambda}. \end{aligned} \tag{2.6}$$

Putting (2.4), (2.5) and (2.6) together,

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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

under the condition

$$\frac{1}{5}\mathbb{E}[Y] - \frac{1}{32} \geqslant \frac{1}{10}\mathbb{E}[Y] + 60\sqrt{\tau},$$

satisfied whenever $\lambda \geqslant 650$. □

3. Partially symmetric functions

A key concept in the study of symmetric and partially symmetric functions is the following notion.

Definition 3.1 ([12, Definition 3.1]). The symmetric influence of a set $T \subseteq [n]$ of variables in a Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is defined as

$$\mathrm{SymInf}_f(T) = \mathbb{P}_{x \in \mathbb{Z}_2^n, \sigma \in S_n}[f(x) \neq f(\sigma(x)) \mid \forall i \notin T : \sigma(i) = i].$$

By definition, a T-symmetric function f has $\mathrm{SymInf}_f(T) = 0$; conversely, for functions far from being T-symmetric we have the following lemma.

Lemma 3.2 ([12, Lemma 3.3]). *If f is ϵ-far from being T-symmetric, then* $\mathrm{SymInf}_f(T) \geqslant \epsilon$. □

In other words, distinguishing between a T-symmetric function and one far from being T-symmetric can be done by estimating the symmetric influence.

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 3.3. $Q^p(\mathrm{Sym}_n) = \Theta(n^{1/4})$ *and* $Q^a(\mathrm{Sym}_n) = O(1)$. □

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 $\mathrm{Bin}(n, 1/2)$, two random inputs share the same Hamming weight with probability

$$4^{-n}\binom{2n}{n} = (1 + o(1))\sqrt{2/\pi n};$$

having fewer than $\frac{1}{2}n^{1/4}$ random samples yields even a single such pair with probability at most

$$\frac{1}{2}\left(\frac{1}{2}n^{1/4}\right)^2 \cdot \sqrt{2/\pi n} < 1/8.$$

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

On the other hand, among $4(2\pi n)^{1/4}$ random samples, it is not hard to see that the probability of not having such a pair is smaller than, say, $1/7$. (One way to show this fact is by looking for matches between the first and second halves of the samples, assuming the first half has not already yielded such a pair. In this case, with high probability the total measure of the layers in which we have a representative from the first half is at least $(2\pi n)^{-1/4}$ and conditioning on this, the probability that no sample from the second half falls into one of these layers is smaller than e^{-2}.)

By Markov, repeating this $14/\epsilon$ times results in at least $12/\epsilon$ sets without a desired pair with probability at most $1/6$. Therefore, with probability at least $5/6$, we have at least $2/\epsilon$ pairs. By Lemma 3.2, if the function is ϵ-far from being symmetric then each such pair will have different outputs with probability at least ϵ, so we will fail to detect this with probability $(1-\epsilon)^{2/\epsilon} < 1/e^2$. Altogether the success probability exceeds $2/3$.

In the context of active testing, given a sample space of, say, $u = n/\epsilon$ vectors we can easily find $2/\epsilon$ input pairs with the same Hamming weight each, thus testing whether the input function is indeed symmetric can be done using $4/\epsilon$ queries. □

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 symmetric influence by counting the number of such examples among all pairs. This enables us to passively (resp. actively) distinguish between a function that is $\epsilon/2$-close to being symmetric and one that is ϵ-far using $O(\epsilon^{-2}n^{1/4})$ (resp. $O(\epsilon^{-2})$) queries. 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 1.6, which relies only on the size of the family, is not tight. We remedy this by proving the following refined version.

Definition 3.4. 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 ϵ-far.

Proposition 3.5. *Let $\mathcal{P}$ be a family of Boolean functions and let $\epsilon > 0$. Then*

$$\lfloor \log |\mathcal{I}_{2\epsilon}(\mathcal{P})| \rfloor \leqslant Q^{\ell}_{\epsilon}(\mathcal{P}) \leqslant \lceil \tfrac{64}{\epsilon} \ln |\mathcal{I}_{\epsilon/2}(\mathcal{P})| \rceil.$$

□

Proof. A proper learning algorithm for $\mathcal{P}$ is required to return a function from $\mathcal{P}$ that is ϵ-close to the input function. Since functions in $\mathcal{I}_{2\epsilon}(\mathcal{P})$ are 2ϵ-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 $1/2$.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

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

$$q = \lceil (64/\epsilon) \ln |\mathcal{I}_{\epsilon/2}(\mathcal{P})| \rceil$$

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 $\mathrm{Bin}(q, \epsilon/2)$ 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 ϵ-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 ϵ-close to f (the obvious candidate being f'). □

Corollary 3.6.

$$Q^{\ell}(\mathrm{Sym}_{n-k}) = \Theta(2^k \sqrt{n-k}) \quad \textit{for } k < n;$$

in particular, $Q^{\ell}(\mathrm{Sym}_n) = \Theta(\sqrt{n})$.

Proof. First, we show that

$$|\mathcal{I}_{\epsilon/2}(\mathrm{Sym}_{n-k})| = 2^{O(2^k \sqrt{n-k})}.$$

The binomial distribution $\mathrm{Bin}(n-k, 1/2)$ is concentrated around its centre, and in particular the middle

$$\ell = 1 + 2\lceil \sqrt{(n-k)\ln(4/\epsilon)/2} \rceil$$

layers account for at least $1 - \epsilon/2$ of the weight. In other words, every $(n-k)$-symmetric function is $(\epsilon/2)$-close to an ℓ-canonical $(n-k)$-symmetric function, which is zero outside the middle ℓ layers. We can thus bound $|\mathcal{I}_{\epsilon/2}(\mathrm{Sym}_{n-k})|$ from above by $2^{2^k \ell}$, the number of ℓ-canonical functions.

For the lower bound, consider the middle $\ell' = 1 + 2\lfloor \sqrt{n-k} \rfloor$ layers. The weight ratio between any pair of these layers is bounded by

$$\binom{n-k}{\lfloor (n-k)/2 \rfloor} \Big/ \binom{n-k}{\lfloor (n-k)/2 - \sqrt{n-k} \rfloor} < e^2.$$

Let $\mathcal{C} \subset \mathbb{Z}_2^{2^k \ell'}$ be an error correcting code of rate $1/2$ and relative distance $1/10$; in other words, $\mathcal{C}$ has at least $2^{2^{k-1}\ell'}$ codewords, every pair of which are $(1/10)$-far. We can interpret each codeword as an ℓ'-canonical $(n-k)$-symmetric function, which is $(1/10e^2)$-far from the rest. Hence we get $|\mathcal{I}_{2\epsilon}(\mathrm{Sym}_{n-k})| \geqslant 2^{2^{k-1}\ell'}$ as long as $\epsilon < 1/20e^2$.

Therefore, for our fixed ϵ, the result follows from Proposition 3.5. □

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Proposition 3.8 provides an upper bound for the query complexity of passive and active testing of partially symmetric functions. Its proof relies on the following simple concentration claim in which we make no attempt to optimize the estimates.

Claim 3.7. *There is an absolute constant $b > 0$ such that for every $c, 0 < c < 1$ the following holds. Let s and t be integers satisfying $s < t$. Let P be an arbitrary probability distribution on t bins, where the probability of each bin is at least c/t. Then, when we throw s balls randomly and independently into t bins according to the probability P, the probability of getting less than $cs^2/9t$ collisions[3] is at most* $\exp(-bcs^2/t)$.

Proof. If the number of occupied bins is less than $s/3$ after $\lceil s/2 \rceil$ balls were thrown, then we already have at least $s/6 > cs^2/9t$ collisions. Otherwise, each of the next $\lfloor s/2 \rfloor$ balls has a probability of at least $cs/3t$ colliding with these occupied bins, independently. The number of collisions created by the last $\lfloor s/2 \rfloor$ balls thus dominates a binomial $\mathrm{Bin}(\lfloor s/2 \rfloor, cs/3t)$ random variable. By Chernoff, it is less than $cs^2/9t$ with probability at most $\exp(-bcs^2/t)$. □

Proposition 3.8.

$$Q^p(\mathrm{Sym}_{n-k}) = O(n^{1/4}2^{k/2}\sqrt{k \log n}) \quad \textit{and} \quad Q^a(\mathrm{Sym}_{n-k}) = O(2^k k \log n),$$

for $k = o(\log n)$. □

Proof. We begin with a passive testing algorithm. Let f be the tested Boolean function. Our algorithm asks $q = d(\epsilon)n^{1/4}2^{k/2}\sqrt{k \log n}$ 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 ϵ-far from being $(n-k)$-symmetric, the algorithm rejects with high probability. Assume this is the case and fix a k-set $T \in \binom{[n]}{k}$ of variables. If we choose a random vector x and another random vector y obtained from x by permuting the elements in $[n] \setminus T$, the probability that $f(x) \neq f(y)$ is at least ϵ by Lemma 3.2. By Claim 3.7 (where each bin corresponds to the ordered pair consisting of the projection on T and the Hamming weight of a typical vector, which is within distance $\Theta(\sqrt{n})$ from $n/2$), for an appropriately chosen $d(\epsilon)$, with probability at least $1 - n^{-k}$ our queries will contain more than

$$0.5\frac{d(\epsilon)^2}{9}k \log n > k \log n/\epsilon$$

random disjoint pairs x, y which have the same Hamming weight and agree on T. The probability that none of these pairs will satisfy $f(x) \neq f(y)$ is at most $(1-\epsilon)^{k \log n/\epsilon} < n^{-k}$. The union bound thus completes the argument.

The same argument implies that the query complexity of active testing is $O(2^k k \log n)$, because the only queries the passive algorithm above actually used are the results for the

$$\Theta(q^2/\sqrt{n}) = \Theta(2^k k \log n)$$

[3] A single collision happens every time we place a ball in an already occupied bin.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

pairs x, y which agree on their Hamming weight. The active algorithm will thus simply select from the sample $\Theta(2^k k \log n)$ disjoint pairs with the same Hamming weight and proceed like the passive algorithm. □

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 3.9. $Q^p(\mathrm{Sym}_{n-k}) = \Omega(n^{1/4}(2^{k/2} + \sqrt{k \log n}))$. □

Proof. As in the proof of Proposition 2.5, 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 $o(n^{1/4}2^{k/2})$.

The second part of the lower bound uses the $\Omega(k \log n)$ bound of Theorem 2.3. 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 $q = o(n^{1/4}\sqrt{k \log n})$ queries. Indeed, assume this many queries were performed and denote by $H \subseteq \{0, 1, \ldots, n\}$ 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. □

Theorem 1.9 follows from Propositions 3.8 and 3.9 and Corollaries 2.4 and 3.6, as well as the results of [12].

4. Low-degree polynomials

We prove Theorem 1.10 for a more general case, allowing $1 \leqslant d \leqslant n^{1/3}$. Let

$$\binom{n}{\leqslant d} = \sum_{i=0}^{d} \binom{n}{i}$$

be the number of monomials of degree at most d. Note that for constant d, we have $\binom{n}{\leqslant d} = \Theta(n^d)$.

Theorem (restatement of Theorem 1.10). $Q^p(\mathrm{Pol}_d) = \Theta(\binom{n}{\leqslant d})$.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Proof. The number of polynomials of degree d is $2^{\binom{n}{\leqslant d}}$, hence by Fact 1.6 and Proposition 1.5, passive testing can be done using $O(\binom{n}{\leqslant d})$ queries. We now show a lower bound of $\Omega(\binom{n}{\leqslant d})$ queries.

Let $x_1, x_2, \ldots, x_q \in \mathbb{Z}_2^n$ be the set of

$$q = \left\lfloor \binom{n}{\leqslant d} / 2e \right\rfloor$$

random queries performed by a passive tester. For $i = 1, \ldots, q$, define $y_i \in \mathbb{Z}_2^{\binom{n}{\leqslant d}}$ 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 $\binom{n}{\leqslant d}$ monomials serve as a basis to Pol_d, $\{y_i\}_{i=1}^q$ being linearly independent implies that every possible output $(f(x_1), \ldots, f(x_q)) \in \mathbb{Z}_2^q$ is equally likely when choosing a random $f \in \mathrm{Pol}_d$, 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 $y_1, \ldots, y_{i-1}$, and then apply the union bound to show that none of these events is likely to occur. Let $V_i = \mathrm{span}\{y_1, \ldots, y_{i-1}\}$ be the linear space spanned by the first $i-1$ vectors. By Lemma 4 from [6], since

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

no more than $2^{\lceil n(1-1/d) \rceil}$ d-evaluations of vectors from $\mathbb{Z}_2^n$ reside in V_i. Thus,

$$\mathbb{P}[y_i \in V_i] \leqslant 2^{-\lceil n/d \rceil}$$

and, by the union bound,

$$\mathbb{P}[\exists i : y_i \in V_i] \leqslant q \cdot 2^{-\lceil n/d \rceil} = o(1) \quad \text{for } d \leqslant n^{1/3}. \qquad \square$$

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 Pol_1 to denote the family of linear functions, even though degree 1 polynomials include both linear and affine functions.

Proposition 4.1. $Q^p(\mathrm{Pol}_1) = n + \Theta(1)$. $\square$

Proof. As in the proof of Theorem 1.10, a linearly independent query set is useless for the testing algorithm. Let $x_1, x_2, \ldots, x_q$ be a sequence of $q \leqslant n$ queries and define X_i to be the event that $x_i \in \mathrm{span}\{x_1, \ldots, x_{i-1}\}$. The probability that some linear dependency exists among the q queries is

$$\mathbb{P}\left[\bigcup_{i=1}^{q} X_i\right] = \mathbb{P}\left[\bigcup_{i=1}^{q} \left(X_i \setminus \bigcup_{j=1}^{i-1} X_j\right)\right] = \sum_{i=1}^{q} \mathbb{P}\left[X_i \setminus \bigcup_{j=1}^{i-1} X_j\right] \leqslant \sum_{i=1}^{q} 2^{i-1-n} = \frac{2^q - 1}{2^n}.$$

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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

Given the computation above, a set of $q \leqslant n-2$ 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 ϵ-far from linear, each additional query is inconsistent with the learned function with a constant probability. □

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, Coppersmith, Håstad, Kiwi and Sudan [5].

Lemma 4.2. *Given a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ that is ϵ-far from being linear,*

$$\mathbb{P}_{x_1,x_2,\ldots,x_{2k} \in \mathbb{Z}_2^n}[f(x_1) + \cdots + f(x_{2k}) = f(x_1 + \cdots + x_{2k})] \leqslant \frac{1}{2} + \frac{1}{2}(1-2\epsilon)^{2k-1}.$$

Proof. Since f is ϵ-far from being linear, when writing it in the Fourier basis

$$f(y) = \sum_{S \subseteq [n]} \hat{f}(S) \sum_{i \in S} y_i,$$

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

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

where the middle equality holds by Parseval's theorem. □

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 1.11). *$Q^a(\mathrm{Pol}_1, u) = \Theta(n/\log u)$, for $u \geqslant n^2$.*

Proof. As in the previous proof, 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 at most q is at most

$$\sum_{i=0}^{q} \binom{u}{i} 2^{-n} \leqslant u^q 2^{-n} = 2^{q \log u - n} \leqslant 2^{-n/2},$$

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

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

Given an input function that is ϵ-far from being linear, we use the test of Lemma 4.2 to identify this. Fix $q = 4\lceil n/\log u\rceil$. Given a sample U of u vectors, it contains $\binom{u}{q/2} > 2^n$ subsets of size $q/2$. By the pigeonhole principle two of these sets have the same sum, hence there is a linear dependency of length at most q. On the other hand, by the previous computation, with high probability there is no linear dependency of size less than $n/(2\log u) = q/8$ hence the length exceeds $q/8$. By Lemma 4.2 the probability that f passes a single such test is at most

$$\frac{1}{2} + \frac{1}{2}(1-2\epsilon)^{q/8-1} < \frac{1}{2} + \frac{1}{2}(1-2\epsilon)^{n/2\log u}.$$

Since ϵ is constant, for large enough n this is smaller than 0.9; therefore 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). □

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 number of queries needed for testing k-linear functions in these new models.

A practical aspect of property testing algorithms that we have not covered is the actual run-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 run-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 this. 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 behaviour 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 2.9, used in the proof of Theorem 2.3, 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 $(\log N)^{1/3} \leqslant k \leqslant (\log N)^{1/2-\delta}$, 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

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

graphs (see [22] for some related results), even the above statement for the dense case is interesting.

References

[1] Alon, N., Kaufman, T., Krivelevich, M., Litsyn, S. and Ron, D. (2003) Testing low-degree polynomials over GF(2). In *Proc. 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.

[2] Alon, N. and Spencer, J. H. (2008) *The Probabilistic Method*, third edition, Wiley.

[3] Babai, L., Fortnow, L. and Lund, C. (1991) Nondeterministic exponential time has two-prover interactive protocols. *Comput. Complexity* **1** 3–40.

[4] Balcan, M.-F., Blais, E., Blum, A. and Yang, L. (2012) Active property testing. In *Proc. 53rd Annual IEEE Symposium on Foundations of Computer Science: FOCS '12*, pp. 21–30.

[5] Bellare, M., Coppersmith, D., Håstad, J., Kiwi, M. and Sudan, M. (1996) Linearity testing in characteristic two. *IEEE Trans. Inform. Theory* **42** 1781–1796.

[6] Ben-Eliezer, I., Hod, R. and Lovett, S. (2012) Random low-degree polynomials are hard to approximate. *Comput. Complexity* **21** 63–81.

[7] Bhattacharyya, A., Kopparty, S., Schoenebeck, G., Sudan, M. and Zuckerman, D. (2010) Optimal testing of Reed–Muller codes. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science: FOCS '10*, pp. 488–497.

[8] Blais, E. (2008) Improved bounds for testing juntas. In *Proc. 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.

[9] Blais, E. (2009) Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing: STOC '09*, pp. 151–158.

[10] Blais, E., Brody, J. and Matulef, K. (2012) Property testing lower bounds via communication complexity. *Comput. Complexity* **21** 311–358.

[11] Blais, E. and Kane, D. (2012) Tight bounds for testing k-linearity. In *Proc. 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.

[12] Blais, E., Weinstein, A. and Yoshida, Y. (2012) Partially symmetric functions are efficiently isomorphism-testable. In *Proc. 53rd Annual IEEE Symposium on Foundations of Computer Science: FOCS '12*, pp. 551–560. Also *SIAM J. Computing* **44** (2015) 411–432.

[13] Blum, M., Luby, M. and Rubinfeld, R. (1993) Self-testing/correcting with applications to numerical problems. In *J. Comput. System Sci.* **47** 549–595.

[14] Chakraborty, S., Fischer, E., García–Soriano, D. and Matsliah, A. (2012) Junto-symmetric functions, hypergraph isomorphism, and crunching. In *Proc. 27th Annual IEEE Conference on Computational Complexity: CCC '12*, pp. 148–158.

[15] Chockler, H. and Gutfreund, D. (2004) A lower bound for testing juntas. *Inform. Process. Lett* **90** 301–305.

[16] Cohn, D., Atlas, L. and Ladner, R. (1994) Improving generalization with active learning. In *Proc. 15th International Conference on Machine Learning: ICML '94*, pp. 201–221.

[17] Diakonikolas, I., Lee, H., Matulef, K., Onak, K., Rubinfeld, R., Servedio, R. and Wan, A. (2007) Testing for concise representations. In *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science: FOCS '07*, pp. 549–558.

[18] Erdős, P. and Rado, R. (1960) Intersection theorems for systems of sets. *J. London Math. Soc.* **35** 85–90.

https://doi.org/10.1017/S0963548315000292 Published online by Cambridge University Press

[19] Fischer, E., Kindler, G., Ron, D., Safra, S. and Samorodnitsky, A. (2004) Testing juntas. *J. Comput. System Sci.* **68** 753–787.
[20] Goldreich, O., Goldwasser, S. and Ron, D. (1998) Property testing and its connection to learning and approximation. *J. Assoc. Comput. Mach.* **45** 653–750.
[21] Kearns, M. and Ron, D. (2000) Testing problems with sublearning sample complexity. *J. Comput. System Sci.* **61** 428–456.
[22] Lubetzky, E. and Sly, A. (2010) Cutoff phenomena for random walks on random regular graphs. *Duke Math. J.***153** 475–510.
[23] Molloy, M. and Reed, B. (2002) *Graph Colouring and the Probabilistic Method*, Springer.
[24] Rubinfeld, R. and Sudan, M. (1996) Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.* **25** 252–271.
[25] Shannon, C. E. (1949) The synthesis of two-terminal switching circuits. *Bell System Tech. J.* **28** 59–98.
[26] Talagrand, M. (1995) Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'IHÉS* **81** 73–203.
[27] Valiant, L. G. (1984) A theory of the learnable. *Comm. Assoc. Comput. Mach.* **27** 1134–1142.