Electronic Colloquium on Computational Complexity, Report No. 123 (2020)

An Optimal Tester for k-Linear

Nader H. Bshouty
Dept. of Computer Science
Technion, Haifa, 32000

August 17, 2020

Abstract

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is k-linear if it returns the sum (over the binary field F_2) of k coordinates of the input. In this paper, we study property testing of the classes k-Linear, the class of all k-linear functions, and k-Linear*, the class $\cup_{j=0}^{k} j$-Linear. We give a non-adaptive distribution-free two-sided ϵ-tester for k-Linear that makes

$$O\left(k\log k + \frac{1}{\epsilon}\right)$$

queries. This matches the lower bound known from the literature.

We then give a non-adaptive distribution-free one-sided ϵ-tester for k-Linear* that makes the same number of queries and show that any non-adaptive uniform-distribution one-sided ϵ-tester for k-Linear must make at least $\tilde{\Omega}(k)\log n + \Omega(1/\epsilon)$ queries. The latter bound, almost matches the upper bound $O(k\log n + 1/\epsilon)$ known from the literature. We then show that any adaptive uniform-distribution one-sided ϵ-tester for k-Linear must make at least $\tilde{\Omega}(\sqrt{k})\log n + \Omega(1/\epsilon)$ queries.

1 Inroduction

Property testing of Boolean function was first considered in the seminal works of Blum, Luby and Rubinfeld [9] and Rubinfeld and Sudan [38] and has recently become a very active research area. See for example, [1, 2, 3, 4, 7, 8, 10, 12, 13, 14, 15, 16, 17, 18, 19, 21, 25, 27, 30, 31, 33, 32, 34, 39] and other works referenced in the surveys and books [23, 24, 35, 36].

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be linear if it returns the sum (over the binary field F_2) of some coordinates of the input, k-linear if it returns the sum of k coordinates, and, k-linear* if it returns the sum of at most k coordinates. The class Linear (resp. k-Linear and k-Linear*) is the classes of all linear functions (resp. all k-linear functions and $\cup_{i=0}^{k} k$-Linear). Those classes has been of particular interest to the property testing community [7, 8, 9, 10, 11, 21, 22, 24, 28, 35, 36, 37, 39].

1.1 The Model

Let f and g be two Boolean functions $\{0,1\}^n \to \{0,1\}$ and let $\mathcal{D}$ be a distribution on $\{0,1\}^n$. We say that f is ϵ-*far* from g with respect to (w.r.t.) $\mathcal{D}$ if $\mathbf{Pr}_{\mathcal{D}}[f(x) \neq g(x)] \geqslant \epsilon$ and ϵ-*close* to g w.r.t. $\mathcal{D}$ if $\mathbf{Pr}_{\mathcal{D}}[f(x) \neq g(x)] \leqslant \epsilon$.

ISSN 1433-8092

In the uniform-distribution and distribution-free property testing model, we consider the problem of testing a class of Boolean function C. In the distribution-free testing model (resp. uniform-distribution testing model), the *tester* is a randomized algorithm that has access to a Boolean function $f : \{0,1\}^n \to \{0,1\}$ via a black-box oracle that returns $f(x)$ when a string x is queried. The tester also has access to unknown distribution $\mathcal{D}$ (resp. uniform distribution) via an oracle that returns $x \in \{0,1\}^n$ chosen randomly according to the distribution $\mathcal{D}$ (resp. according to the uniform distribution). A *distribution-free tester*, [26], (resp. *uniform-distribution tester*) $\mathcal{A}$ for C is an tester that, given as input a distance parameter ϵ and the above two oracles to a Boolean function f,

1. if $f \in C$ then $\mathcal{A}$ accepts with probability at least 2/3.

2. if f is ϵ-far from every $g \in C$ w.r.t. $\mathcal{D}$ (resp. uniform distribution) then $\mathcal{A}$ rejects with probability at least 2/3.

We will also call $\mathcal{A}$ an *ϵ-tester for the class C* or an algorithm for *ϵ-testing C*. We say that $\mathcal{A}$ is *one-sided* if it always accepts when $f \in C$; otherwise, it is called *two-sided* tester. The *query complexity of $\mathcal{A}$* is the maximum number of queries $\mathcal{A}$ makes on any Boolean function f. If the query complexity is q then we call the tester a *q-query tester* or a tester with *query complexity q*.

In the *adaptive testing* (uniform-distribution or distribution-free) the queries can depend on the answers of the previous queries where in the *non-adaptive testing* all the queries are fixed in advance by the tester.

In this paper we study testers for the classes k-Linear and k-Linear*.

1.2 Prior Results

Throughout this paper we assume that $k < \sqrt{n}$. Blum et al. [9] gave an $O(1/\epsilon)$-query non-adaptive uniform-distribution one-sided ϵ-tester (called BLR tester) for Linear. Halevy and Kushilevitz, [28], used a self-corrector (an algorithm that computes $g(x)$ from a black box query to f that is ϵ-close to g) to reduce distribution-free testability to uniform-distribution testability. This reduction gives an $O(1/\epsilon)$-query non-adaptive distribution-free one-sided ϵ-tester for Linear. The reduction can be applied to any subclass of Linear. In particular, any q-query uniform-distribution ϵ-tester for k-Linear (k-Linear*) gives a $O(q)$-query distribution-free ϵ-tester.

It is well known that if there is a q_1-query uniform-distribution ϵ-tester for Linear and a q_2-query uniform-distribution ϵ-tester for the class k-Junta[1] then there is an $O(q_1 + q_2)$-query uniform-distribution $O(\epsilon)$-tester for k-Linear*. Since k-Linear = k-Linear*$\backslash(k-1)$-Linear*, if there is a q-query uniform-distribution ϵ-tester for k-Linear* then there is an $O(q)$-query uniform-distribution two-sided ϵ-tester for k-Linear. Therefore, all the results for testing k-Junta are also true for k-Linear* and k-Linear in the uniform-distribution model.

For lower bounds on the number queries for two-sided uniform-distribution testing k-Linear (see the table in Figure 1): For non-adaptive testers Fisher, et al. [21] gave the lower bound $\Omega(\sqrt{k})$. Goldreich [22], gave the lower bound $\Omega(k)$. In [8], Blais and Kane gave the lower bound $2k - o(k)$. Then in [7], Blais et al. gave the lower bound $\Omega(k \log k)$. For adaptive testers, Goldreich [22], gave the lower bound $\Omega(\sqrt{k})$. Then Blais et al. [7] gave the lower bound $\Omega(k)$ and in [8], Blais and Kane

[1] The class of boolean functions that depends on at most k coordinates

gave the lower bound $k - o(k)$. Then in [39], Saglam gave the lower bound $\Omega(k \log k)$. This bound with the trivial $\Omega(1/\epsilon)$ lower bound gives the lower bound

$$\Omega\left(k \log k + \frac{1}{\epsilon}\right) \tag{1}$$

for the query complexity of any adaptive uniform-distribution (and distribution-free) two-sided testers.

For upper bounds for uniform-distribution two-sided ϵ-testing k-Linear, Fisher, et al. [21] gave the first adaptive tester that makes $O(k^2/\epsilon)$ queries. In [11], Buhrman et al. gave a non-adaptive tester that makes $O(k \log k)$ queries for any constant ϵ. As is mentioned above, testing k-Linear can be done by first testing if the function is k-Junta and then testing if it is Linear. Therefore, using Blais [5, 6] adaptive and non-adaptive testers for k-Junta we get adaptive and non-adaptive uniform-distribution testers for k-Linear that makes $O(k \log k + k/\epsilon)$ and $\tilde{O}(k^{1.5}/\epsilon)$ queries, respectively.

For upper bounds for two-sided distribution-free testing k-Linear, as is mentioned above, from Halevy et al. reduction in [28], an adaptive and non-adaptive distribution-free ϵ-tester can be constructed from adaptive and non-adaptive uniform-distribution ϵ-testers. This gives an adaptive and non-adaptive distribution-free two-sided testers for k-Linear that makes $O(k \log k + k/\epsilon)$ and $\tilde{O}(k^{1.5}/\epsilon)$ queries, respectively. See the table in Figure 1.

1.3 Our Results

In this paper we prove

Theorem 1. *For any $\epsilon > 0$, there is a polynomial time non-adaptive distribution-free one-sided ϵ-tester for k-Linear* that makes*

$$O\left(k \log k + \frac{1}{\epsilon}\right)$$

queries.

By the reduction from k-Linear to k-Linear*, we get

Theorem 2. *For any $\epsilon > 0$, there is a polynomial time non-adaptive distribution-free two-sided ϵ-tester for k-Linear that makes*

$$O\left(k \log k + \frac{1}{\epsilon}\right)$$

queries.

For one-sided testers for k-Linear we prove

Theorem 3. *Any non-adaptive uniform-distribution one-sided ϵ-tester for k-Linear must make at least $\tilde{\Omega}(k) \log n + \Omega(1/\epsilon)$ queries.*

This almost matches the upper bound $O(k \log n + 1/\epsilon)$ that follows from the reduction of Goldreich et. al [26] and the non-adaptive deterministic exact learning algorithm of Hofmeister [29] that learns k-Linear with $O(k \log n)$ queries.

For adaptive testers we prove

Theorem 4. *Any adaptive uniform-distribution one-sided ϵ-tester for k-Linear must make at least $\tilde{\Omega}(\sqrt{k}) \log n + \Omega(1/\epsilon)$ queries.*

The table in 1 summarizes all the results in the literature and our results for the class k-Linear.

Upper/ Lower	One-Sided/ Two-Sided	Adaptive/ Non-Adap.	Uniform/ Dist. Free	Result O/Ω	Reference
Upper	Two-Sided	Adaptive	Uniform	k^2/ϵ	[21]
Upper	Two-Sided	Adaptive	Uniform	$k\log k + k/\epsilon$	[6]
Upper	Two-Sided	Adaptive	Dist. Free	$k\log k + k/\epsilon$	[28]
Upper	Two-Sided	Non-Adap.	Uniform	$k\log k$ (ϵ Const.)	[11]
Upper	Two-Sided	Non-Adap.	Uniform	$k^{1.5}/\epsilon$	[5]
Upper	Two-Sided	Non-Adap.	Dist. Free	$k^{1.5}/\epsilon$	[28]
Upper	Two-Sided	Non-Adap.	Dist. Free	$k\log k + 1/\epsilon$	Ours
Lower	Two-Sided	Non-Adap.	Uniform	$1/\epsilon$	Trivial
Lower	Two-Sided	Non-Adap.	Uniform	$\sqrt{k} + 1/\epsilon$	[21]
Lower	Two-Sided	Non-Adap.	Uniform	$k + 1/\epsilon$	[22]
Lower	Two-Sided	Non-Adap.	Uniform	$k\log k + 1/\epsilon$	[7]
Lower	Two-Sided	Adaptive	Uniform	$\sqrt{k} + 1/\epsilon$	[22]
Lower	Two-Sided	Adaptive	Uniform	$k + 1/\epsilon$	[7, 8]
Lower	Two-Sided	Adaptive	Uniform	$k\log k + 1/\epsilon$	[39]
Upper	One-Sided	Non-Adaptive	Dist. Free	$k\log n + 1/\epsilon$	[26]
Lower	One-Sided	Non-Adaptive	Uniform	$\tilde{\Omega}(k)\log n + 1/\epsilon$	Ours
Lower	One-Sided	Adaptive	Uniform	$\tilde{\Omega}(\sqrt{k})\log n + 1/\epsilon$	Ours

Figure 1: A table of results for the testability of the class k-Linear.

2 Overview of the Testers and Lower Bounds

In this section we give overview of the techniques used for proving the results in this paper.

2.1 One-sided Tester for k-Linear*

The tester for k-Linear* first runs the tester BLR of Blum et al. [9] to test if the function f is ϵ'-close to Linear w.r.t. the uniform distribution, where $\epsilon' = \Theta(1/(k\log k))$. BLR is one-sided tester and therefore, if f is k-linear then BRG accepts with probability 1. If f is ϵ'-far from Linear w.r.t. the uniform distribution then, with probability at least 2/3, BLR rejects. Therefore, if the tester BLR accepts, we may assume that f is ϵ'-close to Linear w.r.t. the uniform distribution. Let $g \in$Linear be the function that is ϵ'-close to f. If f is k-linear* then $f = g$. This is because $\epsilon' < 1/8$ and the distance (w.r.t. the uniform distribution) between every two linear functions is 1/2. BLR makes $O(1/\epsilon') = O(k\log k)$ queries.

In the second stage, the tester tests if g (not f) is k-linear*. Let us assume for now that we can query g in every string. Since $g \in$Linear, we need to distinguish between functions in k-Linear* and functions in Linear$\backslash k$-Linear*. We do that with two tests. We first test if $g \in 8k$-Linear* and then test if it is in k-Linear* assuming that it is in $8k$-Linear*. In the first test, the tester "throws", uniformly at random, the variables of g into $16k$ bins and tests if there is more than k non-empty bins. If g is k-linear* then the number of non-empty bins is always less than k. If it is k'-linear for some $k' > 8k$ then with high probability (w.h.p.) the number of non-empty bins is greater than k. Notice that if f is k-linear* then the test always accepts and therefore it is one-sided. This tests makes $O(k)$ queries to g.

The second test is testing if g is in k-Linear* assuming that it is in $8k$-Linear*. This is done by projecting the variables of g into $r = O(k^2)$ coordinates uniformly at random and learning (finding exactly) the projected function using the non-adaptive deterministic Hofmeister's algorithm, [29], that makes $O(k \log r) = O(k \log k)$ queries. Since $g \in 8k$-Linear*, w.h.p., the relevant coordinates of the function are projected to different coordinates, and therefore, w.h.p., the learning gives a linear function that has exactly the same number of relevant coordinates as g. The tester accepts if the number of relevant coordinates in the projected function is at most k. If $g \in k$-Linear*, then the projected function is in k-Linear* with probability 1 and therefore this test is one-sided. This test makes $O(k \log k)$ queries.

We assumed that we can query g. We now show how to query g in $O(k \log k)$ strings so we can apply the above two tests. For this, the tester uses self-corrector, [9]. To compute $g(z)$, the self-corrector chooses a uniform random string $a \in \{0,1\}^n$ and computes $f(z+a) + f(a)$. Since f is $O(1/(k \log k))$-close to g w.r.t. the uniform distribution, we have that for any string $z \in \{0,1\}^n$ and an $a \in \{0,1\}^n$ chosen uniformly at random, with probability at least $1 - O(1/(k \log k))$, $f(z+a) + f(a) = g(z+a) + g(a) = g(z)$. Therefore, w.h.p., the self-corrector computes correctly the values of g in $O(k \log k)$ strings. If $f \in k$-Linear then $g = f$ and $f(z+a) + f(z) = f(z) = g(z)$, i.e., the self-corrector gives the value of g with probability 1. This shows that the above two tests are one-sided.

Now, if f is k-linear* then $f = g$. If f is ϵ-far from every function in k-Linear* w.r.t. $\mathcal{D}$ then it is ϵ-far from g w.r.t. $\mathcal{D}$.

In the final stage the tester tests whether f is equal to g or ϵ-far from g w.r.t. $\mathcal{D}$. Here again the tester uses self-corrector. It asks for a sample $\{(z^{(i)}, f(z_i)) | i \in [t]\}$ according to the distribution $\mathcal{D}$ of size $t = O(1/\epsilon)$ and tests if $f(z^{(i)}) = f(z^{(i)} + a^{(i)}) + f(a^{(i)})$ for every $i \in [t]$, where $a^{(i)}$ are i.i.d. uniform random strings. If $f(z^{(i)}) = f(z^{(i)} + a^{(i)}) + f(a^{(i)})$ for all i then it accepts, otherwise, it rejects. If f is k-linear then $f(z^{(i)}) = f(z^{(i)} + a^{(i)}) + f(a^{(i)})$ for all i and the tester accepts with probability 1. Now suppose f is ϵ-far from g w.r.t. $\mathcal{D}$. Since f is ϵ'-close to g w.r.t. the uniform distribution and $\epsilon' \leqslant 1/8$ we have that, with probability at least $7/8$, $f(z^{(i)} + a^{(i)}) + f(a^{(i)}) = g(z^{(i)} + a^{(i)}) + g(a^{(i)}) = g(z^{(i)})$. Therefore, assuming the latter happens, then, with probability at least $1 - \epsilon$ we have $f(z^{(i)}) \neq g(z^{(i)}) = f(z^{(i)} + a^{(i)}) + f(a^{(i)})$. Thus, w.h.p, there is i such that $f(z^{(i)}) \neq f(z^{(i)} + a^{(i)}) + f(a^{(i)})$ and the tester rejects. This stage is one-sided and makes $O(1/\epsilon)$ queries.

2.2 Two-sided Testers for k-Linear

As we mentioned in the introduction, the one-sided q-query uniform-distribution ϵ-tester for k-Linear* gives a two-sided uniform-distribution $O(q)$-query ϵ-tester for k-Linear. This is because, in the uniform distribution, the linear functions are 1/2-far from each other and therefore, for any $\epsilon < 1/4$, if f is ϵ-close to a k-linear function g then it is $(1/2 - \epsilon)$-far from $(k-1)$-Linear*. This is not true for any distribution $\mathcal{D}$, and therefore, cannot be applied here.

The algorithm in the previous subsection can be changed to a two-sided tester for k-Linear as follows. The only part that should be changed is the test that g is in k-Linear* assuming that it is in $8k$-Linear*. We replace it with a test that g is in k-Linear assuming that it is in $8k$-Linear*. The tester rejects if the number of relevant coordinates in the function that is learned is not *equal* to k. This time the test is two-sided. The reason is that the projection to $O(k^2)$ variables does not guarantee (with probability 1) that all the variables of f are projected to different variables.

Therefore, it may happen that f is k-linear and the projection gives a $(k-1)$-linear* function.

2.3 The Lower Bound for One-sided Testers

We first show the result for non-adaptive testers. Suppose there is a one-sided non-adaptive uniform distribution 1/8-tester $A(s,f)$ for k-Linear that makes q queries, where s is the random seed of the tester and f is the function that is tested. The algorithm has access to f through a black box queries.

Consider the set of linear functions $C = \{g^{(0)}\} \cup \{g^{(\ell)} = x_n + \cdots + x_{n-\ell+1} | \ell = 1,\ldots,k-1\} \subseteq (k-1)$-Linear* where $g^{(0)} = 0$. Any k-linear function is 1/2-far from every function in C w.r.t. the uniform distribution. Therefore, using the tester A, with probability at least 2/3, we can distinguish between any k-linear and any function in C. By running the tester A $O(\log k)$ times, and accept if and only if all accept, we get a tester A' that asks $O(q \log k)$ queries and satisfies

1. If $f \in k$-Linear then with probability 1, $A'(s,f)$ accepts.
2. If $f \in C$ then, with probability at least $1 - 1/(2k)$, $A'(s,f)$ rejects.

By an averaging argument (i.e., fixing coins for A') and since $|C| = k$, there exists a deterministic non-adaptive algorithm B that makes $q' = O(q \log k)$ queries such that

1. If $f \in k$-Linear then $B(f)$ accepts.
2. If $f = C$ then $B(f)$ rejects.

Let $a^{(i)}$, $i = 1,\ldots,q'$ be the queries that B makes. Let M be a $q' \times n$ binary matrix where the i-th row of M is $a^{(i)}$ and $x^f \in \{0,1\}^n$ where $x_i^f = 1$ if i is a relevant coordinate in f. Then the vector of answers to the queries of $B(f)$ is Mx^f. If $Mx^f = Mx^g$ for some $g \in C$, that is, the answers of the queries to f are the same as the answer of the queries to g, then $B(f)$ rejects. Therefore, for every $f \in k$-Linear and every $g \in C$ we have $Mx^f \neq Mx^g$. Now since $\{x^f | f \in k\text{–Linear}\}$ is the set of all strings of weight k, the sum (over the field F_2) of every k columns of M is not equal to 0 and not equal to the sum of the last ℓ columns of M, for all $\ell = 1,\ldots,k-1$. In particular, if M_i is the ith column of M, for every $i_1,\ldots,i_{k-\ell} \leqslant n-k+1$, $M_{i_1} + \cdots + M_{i_{k-\ell}} + M_{n-\ell+1} + \cdots + M_n \neq M_{n-\ell+1} + \cdots + M_n$ and therefore $M_{i_1} + \cdots + M_{i_{k-\ell}} \neq 0$. That is, the sum of every less or equal $k-1$ columns of the first $n-k+1$ columns of M is not equal to zero. We then show (via Hamming's bound in coding theory) that such matrix has at least $q' = \Omega(k \log n)$ rows. This implies that $q = \Omega((k/\log k)\log n)$. See more details in Subsection 4.1.

For the lower bound for adaptive testers we take $C = \{g^{(\ell)}\}$ for some $\ell \in \{0,1,\ldots,k-1\}$ and get a $q \times n$ matrix M that the sum of every $k-\ell$ columns of M is not zero. We then show, that there exists $\ell \leqslant k-1$ where such a matrix must have at least $q = \tilde{\Omega}(\sqrt{k}\log n)$ rows. See more details in Subsections 4.2 and 4.3.

3 The Testers for k-Linear* and k-Linear

In this section we give the non-adaptive distribution-free one-sided tester for k-Linear* and the non-adaptive distribution-free two-sided tester for k-Linear.

3.1 Notations

In this subsection, we give some notations that we use throughout the paper.

Denote $[n] = \{1, 2, \ldots, n\}$. For $S \subseteq [n]$ and $x = (x_1, \ldots, x_n)$. For $X \subset [n]$ we denote by $\{0,1\}^X$ the set of all binary strings of length $|X|$ with coordinates indexed by $i \in X$. For $x \in \{0,1\}^n$ and $X \subseteq [n]$ we write $x_X \in \{0,1\}^X$ to denote the projection of x over coordinates in X. We denote by 1_X and 0_X the all-one and all-zero strings in $\{0,1\}^X$, respectively. For a variable x_i and a set X, we denote by $(x_i)_X$ the string x' over coordinates in X where for every $j \in X$, $x'_j = x_i$. For $X_1, X_2 \subseteq [n]$ where $X_1 \cap X_2 = \varnothing$ and $x \in \{0,1\}^{X_1}, y \in \{0,1\}^{X_2}$ we write $x \circ y$ to denote their concatenation, i.e., the string in $\{0,1\}^{X_1 \cup X_2}$ that agrees with x over coordinates in X_1 and agrees with y over coordinates in X_2. For $X \subseteq [n]$ we denote $\overline{X} = [n] \backslash X = \{x \in [n] | x \notin X\}$.

For example, if $n = 7$, $X_1 = \{1, 3, 5\}$, $X_2 = \{2, 7\}$, y_2 is a variable and $z = (z_1, z_2, z_3, z_4, z_5, z_6, z_7) \in \{0,1\}^7$ then $(y_2)_{X_1} \circ z_{X_2} \circ 0_{\overline{X_1 \cup X_2}} = (y_2, z_2, y_2, 0, y_2, 0, z_7)$.

3.2 The Tester

Consider the tester **Test-Linear**$^*_k$ for k-Linear* in Figure 2. The tester uses three procedures. The first is **Self-corrector** that for an input $x \in \{0,1\}^n$ chooses a uniform random $z \in \{0,1\}^n$ and returns $f(x+z) + f(z)$. The procedure **BLR** that is a non-adaptive uniform-distribution one-sided ϵ-tester for Linear. BLR makes c_1/ϵ queries for some constant c_1, [9]. The third procedure is **Hoffmeister's Algorithm** (N, K), a deterministic non-adaptive algorithm that exactly learns K-Linear* over N coordinates from black box queries. Hoffmeister's Algorithm makes $c_2 K \log N$ queries for some constant c_2, [29].

To test k-Linear we use the same tester but change step 11 to:
(11) If the output is not in k-Linear then reject

We call this tester **Test-Linear**$_k$.

3.3 Correctness of the Tester

In this section we prove

Theorem 5. **Test-Linear**$_k$ *is a non-adaptive distribution-free two-sided ϵ-tester for k-Linear that makes*

$$O\left(k \log k + \frac{1}{\epsilon}\right)$$

queries.

Theorem 6. **Test-Linear**$^*_k$ *is a non-adaptive distribution-free one-sided ϵ-tester for k-Linear* that makes*

$$O\left(k \log k + \frac{1}{\epsilon}\right)$$

queries.

Proof. Since there is no stage in the tester that uses the answers of the queries asked in previous ones, the tester is non-adaptive.

In Stage 1 the tester makes $O(1/\epsilon') = O(k \log k)$ queries. In stage 2.1, $O(k)$ queries. In stage 2.2, $O(k \log r) = O(k \log k)$ queries and in stage 3, $O(1/\epsilon)$ queries. Therefore, the query complexity of the tester is $O(k \log k + 1/\epsilon)$.

Test-Linear$_k^*$
Input: Oracle that accesses a Boolean function f
Output: Either "Accept" or "Reject"

Procedures

Self-corrector $g(x) := f(x+z) + f(z)$ for uniform random $z \in \{0,1\}^n$.
BLR A procedure that ϵ-tests Linear using c_1/ϵ queries.
Hofmeister's Algorithm(N, K) for learning K-Linear* over
N coordinates using $c_2 K \log N$ queries.

Stage 1. BLR
1. Run BLR on f with $\epsilon' = 1/(12(16k + c_2 k \log(256k^2)))$
2. If BLR rejects then reject.

Stage 2.1. Testing if g is in Linear$\backslash 8k$-Linear*
3. Choose a uniform random partition $X_1, \ldots, X_{16k}$
4. $Count \leftarrow 0$;
5. Choose a uniform random $z \in \{0,1\}^n$.
6. For $i = 1$ to $16k$
7. if $g(z_{X_i} \circ 0_{\overline{X_i}}) = 1$ then $Count \leftarrow Count + 1$
8. If $Count > k$ then reject.

Stage 2.2. Testing if g is in k-Linear assuming it is in $8k$-Linear*
9. Choose a uniform random partition $X_1, \ldots, X_r$ for $r = 256k^2$
10. Run Hofmeister's algorithm $(N = r, K = 8k)$ in order
to learn $F = g((y_1)_{X_1} \circ (y_2)_{X_2} \circ \cdots \circ (y_r)_{X_r})$
11. If the output is not in k-Linear* then reject
/* In **Test-Linear$_k$** (for testing k-Linear) we replace (11) with:
/* 11. If the output is not in k-Linear then reject

Stage 3. Consistency test
12. Choose a sample $x^{(1)}, \ldots, x^{(t)}$ according to $\mathcal{D}$ of size $t = 4/\epsilon$.
13. For $i = 1$ to t.
14. If $f(x^{(i)}) \neq g(x^{(i)})$ then reject.
15. Accept.

Figure 2: An optimal two-sided tester for k-Linear.

We will assume that $k \geqslant 12$. For $k < 12$, (see the introduction and Table 1) the non-adaptive tester of k-Junta with the BLR tester and the self-corrector gives a non-adaptive testers that makes $O(1/\epsilon) = O(k \log k + 1/\epsilon)$ queries.

Completeness: We first show the completeness for **Test-Linear$_k$** that tests k-Linear. Suppose $f \in k$-Linear. Then for every x we have $g(x) = f(x+z) + f(z) = f(x) + f(z) + f(z) = f(x)$. Therefore, $g = f$. In stage 1, BLR is one-sided and therefore it does not reject. In stage 2.1, since $X_1, \ldots, X_{16k}$ are pairwise disjoint, the number of functions $g(x_{X_i} \circ 0_{\overline{X_i}})$, $i = 1, 2, \ldots, 16k$, that are not identically zero is at most k and therefore stage 2.1 does not reject. In stage 2.2, with probability at least $1-\binom{k}{2}/(256k^2)\geqslant 2/3$, the relevant coordinates of f fall into different X_i and then $F=g((y_1)_{X_1}\circ(y_2)_{X_2}\circ\cdots\circ(y_r)_{X_r})=f((y_1)_{X_1}\circ(y_2)_{X_2}\circ\cdots\circ(y_r)_{X_r})$ is k-linear. Then, Hofmeister's algorithm returns a k-linear function. Therefore, with probability at least 2/3 the tester does not reject. Stage 3 does not reject since $f=g$.

Now for the tester **Test-Linear**$^*_k$, in stage 2.2, with probability 1 the function F is in k-Linear*. In fact, if t relevant coordinates falls into the set X_i then the coordinate i (that correspond to the variable y_i) will be relevant in F if and only if t is odd. Therefore, the tester does not reject.

Notice that **Test-Linear**$^*_k$ is one-sided and **Test-Linear**$_k$ is two-sided.

Soundness: We prove the soundness for **Test-Linear**$_k$. The same proof also works for **Test-Linear**$^*_k$. Suppose f is ϵ-far from k-Linear w.r.t. the distribution $\mathcal{D}$. We have four cases

Case 1 : f is ϵ'-far from Linear w.r.t. the uniform distribution.

Case 2 : f is ϵ'-close to $g\in$Linear and g is in Linear$\backslash 8k$-Linear*.

Case 3 : f is ϵ'-close to $g\in$Linear and g is in $8k$-Linear*$\backslash k$-Linear.

Case 4 : f is ϵ'-close to $g\in$Linear, g is in k-Linear and f is ϵ-far from k-Linear w.r.t. $\mathcal{D}$.

For Case 1, if f is ϵ'-far from Linear then, in stage 1, BLR rejects with probability 2/3.

For Cases 2 and 3, since f is ϵ'-close to g, for any fixed $x\in\{0,1\}^n$ with probability at least $1-2\epsilon'$ (over a uniform random z), $f(x+z)+f(z)=g(x+z)+g(z)=g(x)$. Since stages 2.1 and 2.2 makes $(16k+c_2k\log r)$ queries (to g), with probability at least $1-(16k+c_2k\log r)2\epsilon'\geqslant 5/6$, $g(x)$ is computed correctly for all the queries in stages 2.1 and 2.2.

For Case 2, consider stage 2.1 of the tester. If g is in Linear$\backslash 8k$-Linear* then g has more than $8k$ relevant coordinates. The probability that less than or equal to $4k$ of $X_1,\ldots,X_{16k}$ contains relevant coordinates of g is at most

$$\binom{16k}{4k}\frac{1}{4^{8k}}\leqslant\left(\frac{e16k}{4k}\right)^{4k}\frac{1}{4^{8k}}\leqslant\frac{1}{12}.$$

If X_i contains the relevant coordinates $i_1,\ldots,i_\ell$ then $g(x_{X_i}\circ 0_{\overline{X}_i})=x_{i_1}+\cdots+x_{i_\ell}$ and therefore, for a uniform random $z\in\{0,1\}^n$, with probability at least 1/2, $g(z_{X_i}\circ 0_{\overline{X}_i})=1$. Therefore, if at least $4k$ of $X_1,\ldots,X_{16k}$ contains relevant coordinates then, by Chernoff bound, with probability at least $1-e^{-k/4}\geqslant 11/12$, the counter "*Count*" is greater than k. Therefore, for Case 2, if g is in Linear$\backslash 8k$-Linear* then, with probability at least $1-(1/6+1/12+1/12)=2/3$, the tester rejects.

For Case 3, consider stage 2.2. If g is in $8k$-Linear*$\backslash k$-Linear then g has at most $8k$ relevant coordinates. Then with probability at least $1-\binom{8k}{2}/(256k^2)\geqslant 5/6$, the relevant coordinates of g fall into different X_i and then Hofmeister's algorithm returns a linear function with the same number of relevant coordinates as g. Therefore stage 2.2 rejects with probability at least 2/3.

For Case 4, if g is in k-Linear and f is ϵ-far from k-Linear w.r.t. $\mathcal{D}$, then f is ϵ-far from g w.r.t. $\mathcal{D}$. Then for uniform random z and $x\sim\mathcal{D}$,

$$\begin{aligned}\mathbf{Pr}_{\mathcal{D},z}[f(x)\neq g(x)] &\geqslant \mathbf{Pr}_{\mathcal{D},z}[f(x)\neq g(x)|g(x)=f(x+z)+f(z)]\mathbf{Pr}_{\mathcal{D},z}[g(x)=f(x+z)+f(z)]\\ &= \mathbf{Pr}_{\mathcal{D}}[f(x)\neq g(x)]\mathbf{Pr}_z[g(x)=f(x+z)+f(z)]\\ &\geqslant \epsilon(1-\epsilon')\geqslant\epsilon/2.\end{aligned}$$

Therefore, with probability at most $(1-\epsilon/2)^t=(1-\epsilon/2)^{4/\epsilon}\leqslant 1/3$, stage 3 does not reject. □

4 Lower Bound

In this section we prove

Theorem 7. *Any non-adaptive uniform-distribution one-sided* $1/8$*-tester for* k*-Linear must make at least* $\tilde{\Omega}(k\log n)$ *queries.*

Theorem 8. *Any adaptive uniform-distribution one-sided* $1/8$*-tester for* k*-Linear must make at least* $\tilde{\Omega}(\sqrt{k}\log n)$ *queries.*

4.1 Lower Bound for Non-Adaptive Testers

We first show the result for non-adaptive testers.

Suppose there is a non-adaptive uniform-distribution one-sided $1/8$-tester $A(s,f)$ for k-Linear that makes q queries, where s is the random seed of the tester and f is the function that is tested. The algorithm has access to f through a black box queries.

Consider the set of linear functions $C = \{g^{(0)}\} \cup \{g^{(\ell)} = x_n + \cdots + x_{n-\ell+1} | \ell = 1,\ldots,k-1\} \subseteq (k-1)$-Linear* where $g^{(0)} = 0$. Any k-linear function is $1/2$-far from every function in C w.r.t. the uniform distribution. Therefore, using the tester A, with probability at least $2/3$, A can distinguish between any k-linear function and functions in C. We boost the success probability to $1 - 1/(2k)$ by running A, $\log(2k)/\log 3$ times, and accept if and only if all accept. We get a tester A' that asks $O(q\log k)$ queries and satisfies

1. If $f \in k$-Linear then with probability 1, $A'(s,f)$ accepts.
2. If $f \in C$ then, with probability at least $1 - 1/(2k)$, $A'(s,f)$ rejects.

Therefore, the probability that for a uniform random s, $A'(s,f)$ accepts for some $f \in C$ is at most $1/2$. Thus, there is a seed s_0 such that $A'(s_0,f)$ rejects for all $f \in C$ (and accept for all $f \in k$-Linear). This implies that there exists a deterministic non-adaptive algorithm $B(= A'(s_0,*))$ that makes $q' = O(q\log k)$ queries such that

1. If $f \in k$-Linear then $B(f)$ accepts.
2. If $f \in C$ then $B(f)$ rejects.

Let $a^{(i)}$, $i = 1,\ldots,q'$ be the queries that B makes. Let M be a $q' \times n$ binary matrix that it's i-th row is $a^{(i)}$. Let $x^f \in \{0,1\}^n$ where $x^f_i = 1$ iff i is relevant coordinate in f. Then the vector of answers to the queries of $B(f)$ is Mx^f. If $Mx^f = Mx^g$ for some $g \in C$, that is, the answers of the queries to f are the same as the answers of the queries to g, then $B(f)$ rejects. Therefore, for every $f \in k$-Linear and every $g \in C$ we have $Mx^f \neq Mx^g$. Now since $\{x^f | f \in k-\text{Linear}\}$ is the set of all strings of weight k, the sum (over the field F_2) of every k columns of M is not equal to 0 (zero string) and not equal to the sum of the last ℓ columns of M, for all $\ell = 1,\ldots,k-1$. In particular, if M_i is the ith column of M, for every $i_1,\ldots,i_{k-\ell} \leqslant n-k+1$, $M_{i_1} + \cdots + M_{i_{k-\ell}} + M_{n-\ell+1} + \cdots + M_n \neq M_{n-\ell+1} + \cdots + M_n$ and therefore $M_{i_1} + \cdots + M_{i_{k-\ell}} \neq 0$. That is, the sum of every less or equal k columns of the first $n-k+1$ columns of M is not equal to zero. We then show in Lemma 10 that such matrix has at least $q' = \Omega(k\log n)$ rows. This implies that $q = \Omega((k/\log k)\log n)$.

Let $\pi(n,k)$ be the minimum integer q such that there exists a $q \times n$ matrix over F_2 that the sum of any of its less than or equal k columns is not 0. We have proved

Lemma 9. *Any non-adaptive uniform-distribution one-sided* $1/8$*-tester for* k*-Linear must make at least* $\Omega(\pi(n-k+1,k)/\log k)$ *queries.*

Now to show that $\Omega(\pi(n-k+1,k)/\log k) = \Omega(k\log n)$ we prove the following result. This lemma follows from Hamming's bound in coding theory. We give the proof for completeness

Lemma 10. *(Hamming's Bound) We have*

$$\pi(n,k) \geqslant \log \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{n}{i} = \Omega(k\log(n/k)).$$

Proof. Let M be a $\pi(n,k)\times n$ matrix over F_2 that the sum of any of its less than or equal k columns is not 0. Let $m = \lfloor k/2 \rfloor$ and $S = \{M_{i_1} + \cdots + M_{i_t} \mid t \leqslant m \text{ and } 1 \leqslant i_1 < \cdots < i_t \leqslant n\} \subseteq \{0,1\}^{\pi(n,k)}$ be a multiset. The strings in S are distinct because, if for the contrary, we have two strings in S that satisfies $M_{i_1} + \cdots + M_{i_t} = M_{j_1} + \cdots + M_{j_{t'}}$ then $M_{i_1} + \cdots + M_{i_t} + M_{j_1} + \cdots + M_{j_{t'}} = 0$ (equal columns are cancelled) and $t + t' \leqslant k$, which is a contradiction. Therefore, $2^{\pi(n,k)} \geqslant |S| = \sum_{i=0}^{m} \binom{n}{i}$ and $\pi(n,k) \geqslant \log |S|$. □

4.2 Lower Bound for Adaptive Testers

For the lower bound for adaptive testers we take $C = \{g^{(\ell)}\}$ for some $\ell \in \{0,1,\ldots,k-1\}$ and get an adaptive algorithm A that makes q queries and satisfies

1. If $f \in k$-Linear then with probability 1, $A(s,f)$ accepts.
2. If $f = g^{(\ell)}$ then, with probability at least $2/3$, $A(s,f)$ rejects.

This implies that there exists a deterministic adaptive algorithm $B = A(s_0, *)$ that makes q queries such that

1. If $f \in k$-Linear then $B(f)$ accepts.
2. If $f = g^{(\ell)}$ then $B(f)$ rejects.

Then, by the same argument as in the case of non-adaptive tester, we get a $q\times n$ matrix M that the sum of every $k-\ell$ columns of the first $n-\ell$ columns of M is not zero. Let $\Pi(n,k)$ be the minimum integer q such that there exists a $q \times n$ matrix over F_2 that the sum of any of its k columns is not 0. Then, we have proved that

Lemma 11. *Any adaptive uniform-distribution one-sided* $1/8$*-tester for* k*-Linear must make at least* $\Omega(\max_{1\leqslant \ell \leqslant k} \Pi(n-k,\ell))$ *queries.*

In the next subsection, we show that there exists $1 \leqslant \ell \leqslant k$ such that $\Pi(n,\ell) = \tilde{\Omega}(\sqrt{k}\log n)$.

4.3 A Lower Bound for Π

In this section we prove

Lemma 12. *We have* $\max_{1\leqslant \ell \leqslant k} \Pi(n,\ell) = \tilde{\Omega}(\sqrt{k}\log n)$.

The idea of the proof is the following. For a set of integers L an L-*good matrix* M is a matrix that for every $\ell \in L$ the sum of every ℓ columns of M is not zero. A k-good matrix is a $\{k\}$-good matrix. We say that the matrix M is *almost L-good* if there is a "small" number ($poly(k)$) of columns of M that can be removed to get an L-good matrix. The concatenation $M_1 \circ M_2$ (the matrix that contains the rows of both matrices) of almost L_1-good matrix M_1 with an almost L_2-good matrix M_2 is an almost $L_1 \cup L_2$-good matrix.

Let $K = \lfloor \sqrt{k}/(2\log k)\rfloor$ and $[K] = \{1, 2, \ldots, K\}$. The idea of the proof is to construct an almost $[K]$-good matrix M by concatenating $t = O(\log k)$ matrices $M_1 \circ M_2 \circ \cdots \circ M_t$ where M_i is k_i-good $(\Pi(n, k_i) \times n)$-matrices for some $k_i \leqslant k$. Then after removing small number ($poly(k)$) columns of M we get a $[K]$-good matrix M with $\sum_{i=1}^{t} \Pi(n, k_i)$ rows and $n - poly(k)$ columns. By Hamming's bound, Lemma 10, M contains at least $\Omega(K \log n)$ rows. Therefore, $\sum_{i=1}^{t} \Pi(n, k_i) = \Omega(K \log n)$. So there is i such that $\Pi(n, k_i) = \Omega(K \log n/ \log k) = \Omega(\sqrt{k} \log n/ \log^2 k) = \tilde{\Omega}(\sqrt{k} \log n)$.

We now give more intuition to how to construct an almost $[K]$-good matrix from k_i-good matrices. Denote by $\mathbb{N}_d = \{i : d \nmid i\} \cap [K]$. Let $k = k_1$. We first show that if M_1 is k_1-good matrix then there exists a set of integers $L_1 \subseteq [K]$ such that M_1 is almost L_1-good matrix and $d_1 := \gcd([K] \backslash L_1) \nmid k_1$. The intuition is that if, for the contrary, there are many pairwise disjoint sets of columns that sum to 0 that the great common divisor of their sizes divides k_1, then the union of some of them gives k_1-set of columns that sum to 0 and then we get a contradiction. Therefore $d_1 \neq 1$, $L_1 \supseteq \mathbb{N}_{d_1}$ and M_1 is almost $\mathbb{N}_{d_1}$-good. We then take $k_2 := d_1 \lfloor k/d_1 \rfloor$ and a k_2-good $\Pi(n, k_2) \times n$ matrix M_2. Then, as before, M_2 is almost $\mathbb{N}_{d_2}$-good matrix with $d_2 \nmid k_2$. Therefore, $d_2 \nmid d_1$. Now the concatenation of both matrices $M_1 \circ M_2$ is almost $\mathbb{N}_{d_1} \cup \mathbb{N}_{d_2} = \mathbb{N}_{\mathrm{lcm}(d_1,d_2)}$. Since $d_2 \nmid d_1$ we must have $d_2' := \mathrm{lcm}(d_1, d_2) \geqslant 2d_1$. We then take $k_3 = d_2' \lfloor k/d_2' \rfloor$ and a k_3-good $\Pi(n, k_3) \times n$ matrix M_3 and concatenate it with $M_1 \circ M_2$ to get an almost $\mathbb{N}_{\mathrm{lcm}(d_1,d_2,d_3)}$-good matrix with $\mathrm{lcm}(d_1, d_2, d_3) \geqslant 2d_2' = 2\,\mathrm{lcm}(d_1, d_2) \geqslant 4d_1$. After, $t = O(\log k)$ iterations, we get a $(\sum_{i=1}^{t} \Pi(n, k_i)) \times n$ matrix $M = M_1 \circ M_2 \circ \cdots \circ M_t$ that is almost $\mathbb{N}_d$-good for some $d \geqslant 2^t d_1 > K$ and therefore, M is almost $[K]$-good.

We note here that we can get the bound $\Omega(\sqrt{k}(\log \log k) \log n/ \log^2 k)$ by choosing $k_1 = \mathrm{lcm}(1, 2, 3, \cdots, m_i) \leqslant k$, and then $k_i = d_{i-1}' \,\mathrm{lcm}(1, 2, 3, \cdots, m_i) < k$ where $m_i = O(\log(k))$. See [20].

We now give the full proof. We start with some preliminary results, Lemmas 13-18.

Lemma 13. *Let $W \subseteq [m]$ and $w = \gcd(W)$. There is a subset $W' \subseteq W$ of size*

$$O\left(\frac{\log \frac{m}{w}}{\log\log \frac{m}{w}}\right) < \log \frac{m}{w}$$

such that $\gcd(W') = \gcd(W)$.

Proof. Define the set $D = W/w = \{b/w | b \in W\}$. Then $D \subseteq [\lfloor m/w \rfloor]$ and $\gcd(D) = 1$. Let $D' \subseteq D$ be a minimum size set with $\gcd(D') = 1$ and $W' = wD' \subseteq W$. Let $D' = \{d_1, \ldots, d_t\}$ and $g_i = \gcd(D' \backslash \{d_i\})$ for $i = 1, \ldots, t$. Since D' is minimum $g_i > 1$. We also have for $i \neq j$,

$$1 = \gcd(D') = \gcd(\gcd(D' \backslash \{d_i\}), \gcd(D' \backslash \{d_j\})) = \gcd(g_i, g_j)$$

and therefore $g_1, \ldots, g_t$ are pairwise relatively prime. Since for all $i > 1$, $g_i = \gcd(D' \backslash \{d_i\}) | d_1$ we have $\prod_{i=2}^{t} g_i | d_1$. Therefore, $\lfloor m/w \rfloor \geqslant d_1 \geqslant \prod_{i=2}^{t} g_i \geqslant \prod_{i=2}^{t} i = t! = |D'|! = |W'|!$ and the result follows. $\square$

Lemma 14. *Let $d, d', k, y \geqslant 1$ be integers that satisfy $d|y, d|k, d|d'$ and $\gcd(y, d') = d$. There is $0 \leqslant \lambda < d'/d$ such that $d' | (k - \lambda y)$.*

Proof. Let $\hat{y} = y/d, \hat{k} = k/d$ and $\hat{d} = d'/d$. Then $\gcd(\hat{y}, \hat{d}) = 1$. Consider the set $B = \{\hat{k} - i\hat{y} \mid i = 0, \ldots, \hat{d}-1\}$. If for $0 \leqslant i_1 < i_2 \leqslant \hat{d}-1$ we have $\hat{k} - i_1\hat{y} = (\hat{k} - i_2\hat{y} \mod \hat{d})$ then $(i_1 - i_2)\hat{y} = (0 \mod \hat{d})$. Since $\gcd(\hat{y}, \hat{d}) = 1$ we get $i_1 = (i_2 \mod \hat{d})$ and therefore $i_1 = i_2$. This shows that the elements in B are distinct modulo $\hat{d}$ and therefore there is $0 \leqslant \lambda < \hat{d} = d'/d$ such that $\hat{k} - \lambda\hat{y} = (0 \mod \hat{d})$. Then $k - \lambda y = (0 \mod d')$. □

Lemma 15. *Let k be an integer. Let $J = \{j_1, \ldots, j_\ell\}$ be a set of integers such that $1 \leqslant j_1, \ldots, j_\ell \leqslant \sqrt{k}/\ell$ and $d := \gcd(j_1, \ldots, j_\ell)|k$. There exist non-negative integers $0 \leqslant \lambda_1, \ldots, \lambda_{\ell-1} \leqslant \sqrt{k}$ and $0 \leqslant \lambda_\ell \leqslant k$ such that*

$$\lambda_1 j_1 + \lambda_2 j_2 + \cdots + \lambda_\ell j_\ell = k.$$

Proof. We prove the result by induction on ℓ. For $\ell = 1$, given $J = \{j_1\}$, $1 \leqslant j_1 \leqslant \sqrt{k}$ and $d = j_1|k$ we let $\lambda_1 = k/d$. Then $\lambda_1 \leqslant k$ and $\lambda_1 j_1 = k$.

Assume that the result is true for $\ell - 1$. We prove the result for ℓ.

Given $d := \gcd(j_1, \ldots, j_\ell)|k$. Let $d' = \gcd(j_2, \ldots, j_\ell)$. We have two cases: $d' = d$ and $d' > d$. If $d' = d$ then $d'|k$ and for $i > 1$, $j_i \leqslant \sqrt{k}/\ell \leqslant \sqrt{k}/(\ell-1)$. By the induction hypothesis there are $0 \leqslant \lambda_2, \ldots, \lambda_{\ell-1} \leqslant \sqrt{k}$ and $0 \leqslant \lambda_\ell \leqslant k$ such that $\lambda_2 j_2 + \lambda_3 j_3 + \cdots + \lambda_\ell j_\ell = k$. We choose $\lambda_1 = 0$ and the result follows.

Now suppose $d' > d$. We have $d|j_1$, $d|k$, $d|d'$ and $\gcd(j_1, d') = d$. By Lemma 14, there is λ_1 such that $0 \leqslant \lambda_1 < d'/d$ and $d'|k' := k - \lambda_1 j_1$. Since $\lambda_1 < d'/d \leqslant j_2 \leqslant \sqrt{k}$, we also have

$$k' = k - \lambda_1 j_1 \geqslant k - \sqrt{k}\sqrt{k}/\ell = \frac{\ell-1}{\ell}k$$

and therefore $j_2, \ldots, j_\ell \leqslant \sqrt{k}/\ell \leqslant \sqrt{k'}/(\ell-1)$. Since $d'|k'$, by the induction hypothesis there exist $0 \leqslant \lambda_2, \ldots, \lambda_{\ell-1} \leqslant \sqrt{k'} \leqslant \sqrt{k}$ and $0 \leqslant \lambda_\ell \leqslant k' < k$ such that $\lambda_2 j_2 + \lambda_3 j_3 + \cdots + \lambda_\ell j_\ell = k'$. Then $\lambda_1 j_1 + \lambda_2 j_2 + \cdots + \lambda_\ell j_\ell = k$. □

Let M be a $q \times n$ binary matrix. Recall that M_i is the ith column of M. For every $j \geqslant 1$, let $\ell_j(M)$ denotes the maximum number of disjoint j-subsets $A_1, A_2, \ldots$ of $[n]$ such that $\sum_{j \in A_i} M_j = 0$ for all i. We say that M is (j, ℓ)-*good* if $\ell_j(M) \leqslant \ell$ and (j, ℓ)-*bad* if it is not (j, ℓ)-good, i.e., $\ell_j(M) > \ell$. For $L, J \subseteq [n]$, we say that M is (L, ℓ)-good if it is (j, ℓ)-good for all $j \in L$ and (J, ℓ)-bad if it is (g, ℓ)-bad for all $j \in J$. When $\ell = 0$ we just say j-good, L-good, j-bad and J-bad.

For two $q_1 \times n$ and $q_2 \times n$ matrices M and M', respectively, the *concatenation of M and M'* is $M \circ M' = [M^*|M'^*]^*$ where $*$ is the transpose of a matrix. That is, $M \circ M'$ is the $(q_1 + q_2) \times n$ matrix that results from the rows of M follows by the rows of M'.

The following result is obvious

Lemma 16. *If M is (L, ℓ)-good and M' is (L', ℓ)-good then $M \circ M'$ is $(L \cup L', \ell)$-good.*

Lemma 17. *Let M be a $q \times n$ matrix. If M is $([d], \ell)$-good then $q = \Omega(d \log((n - (\ell d^2/2))/d))$.*

Proof. For every $j \in [d]$ we have $\ell_j(M) \leqslant \ell$. That is, for every j, there are at most ℓ disjoint j-sets of columns that sum to zero. We remove those columns (for all $j \in [d]$) and get a $([d], 0)$-good matrix. The number of columns that are removed is at most $\sum_{j=1}^{d} \ell j \leqslant \ell d^2/2$. Using Hamming's bound, Lemma 10, the result follows. □

We now prove

Lemma 18. *Let m, q, w and $t = mqw$ be integers. Let $J = \{j_1, \ldots, j_w\} \subseteq [m]$. Let M be a (J, t)-bad matrix. Then for any $\lambda_1, \ldots, \lambda_w \in [q]$ we have that M is $(\lambda_1 j_1 + \cdots + \lambda_w j_w)$-bad.*

Proof. Let $r = \lambda_1 j_1 + \cdots + \lambda_w j_w$. We need to show that there are r columns of M that sum to 0. Since M is (j_1, t)-bad and $\lambda_1 \leqslant t$, there are λ_1 pairwise disjoint j_1-sets $A_{1,1}, A_{1,2}, \cdots, A_{1,\lambda_1}$ such that $\sum_{j \in A_{1,i}} M_j = 0$ for all $i \in [\lambda_1]$. Since M is (j_2, t)-bad and $\lambda_2 \leqslant t - \lambda_1 j_1$, there are λ_2 pairwise disjoint j_2-sets $A_{2,1}, A_{2,2}, \cdots, A_{2,\lambda_2}$ sets that are also pairwise disjoint with $A_{1,1}, A_{1,2}, \cdots, A_{1,\lambda_1}$ such that $\sum_{j \in A_{2,i}} M_j = 0$ for all $i \in [\lambda_2]$. We continue with this procedure until we find a collection $\mathcal{A}'$ of disjoint sets that contains, for every $i \leqslant w-1$, λ_i j_i-sets that corresponds to columns of M that sum to 0. Now since $\lambda_w \leqslant t - (\lambda_1 j_1 + \cdots + \lambda_{w-1} j_{w-1})$, there are λ_w pairwise disjoint j_w-sets $A_{w,1}, A_{w,2}, \cdots, A_{w,\lambda_w}$ sets that are also pairwise disjoint with all the sets in $\mathcal{A}'$ such that $\sum_{j \in A_{w,i}} M_j = 0$ for all $i \in [\lambda_w]$. Let $\mathcal{A} = \mathcal{A}' \cup \{A_{w,i} | i \in [\lambda_w]\}$. Obviously, $|\cup \mathcal{A}| = \lambda_1 j_1 + \cdots + \lambda_w j_w$ and $\sum_{j \in \cup \mathcal{A}} M_j = 0$. □

We now show that if a k-good matrix M is $(J, poly(k))$-bad then $\gcd(J) \nmid k$.

Lemma 19. *Let $K = \lfloor \sqrt{k}/(2 \log k) \rfloor$, $\kappa = k^{1.5}$, $J \subseteq [K]$ and $k/2 \leqslant k' \leqslant k$. Let M be a matrix that is k'-good and (J, κ)-bad. Then $\gcd(J) \nmid k'$.*

Proof. Let $d = \gcd(J)$ and suppose, for the contrary, that $d | k'$. By Lemma 13, there is $J' \subseteq J$ of size $w := |J'| \leqslant \log(K/d) < \log k$ such that $d = \gcd(J')$. Let $J' = \{j_1, \ldots, j_w\}$. By Lemma 15, there exist $0 \leqslant \lambda_1, \ldots, \lambda_w \leqslant k$ such that $\lambda_1 j_1 + \cdots + \lambda_w j_w = k'$. By Lemma 18, M is k'-bad. A contradiction. □

Let $K = \lfloor \sqrt{k}/(2 \log k) \rfloor$ and $\kappa = k^{1.5}$. Let $\mathbb{N}_d$ be the set of integers in $[K]$ that are not divisible by d.

Lemma 20. *Let J be the maximum subset of $[K]$ such that M is (J, κ)-bad. Then M is $(\mathbb{N}_{\gcd(J)}, \kappa)$-good.*

Proof. Since J is the maximum set, M is $([K] \backslash J, \kappa)$-good. Since $J \subseteq [K] \backslash \mathbb{N}_{\gcd(J)}$ we have $[K] \backslash J \supseteq \mathbb{N}_{\gcd(J)}$ and therefore M is $(\mathbb{N}_{\gcd(J)}, \kappa)$-good. □

We now show how to construct from a $(\mathbb{N}_d, \kappa)$-good matrix a $(\mathbb{N}_{d'}, \kappa)$-good matrix with $d' \geqslant 2d$.

Lemma 21. *Let M be a $q \times n$ matrix that is $(\mathbb{N}_d, \kappa)$-good. There exist $k' \leqslant k$, $q' = q + \Pi(k', n)$, $d' \geqslant 2d$ and a $q' \times n$ matrix M' that is $(\mathbb{N}_{d'}, \kappa)$-good.*

Proof. Consider $k' = d \lfloor k/d \rfloor$ and let $\hat{M}$ be a $\Pi(n, k') \times n$ matrix that is k'-good. Let J' be the maximum subset of $[K]$ such that $\hat{M}$ is (J', κ)-bad. By Lemma 19, $\gcd(J') \nmid k' = d \lfloor k/d \rfloor$ and therefore $\gcd(J') \nmid d$. By Lemma 20, $\hat{M}$ is $(\mathbb{N}_{\gcd(J')}, \kappa)$-good. Define $M' = M \circ \hat{M}$.

First, the number of rows of M' is $q' = q + \Pi(k', n)$. Now, by Lemma 16, M' is $(\mathbb{N}_{\gcd(J')} \cup \mathbb{N}_d, \kappa)$-good. Since $\mathbb{N}_{\gcd(J')} \cup \mathbb{N}_d = \mathbb{N}_{d'}$ for $d' = \mathrm{lcm}(\gcd(J'), d)$ we have that M' is $(\mathbb{N}_{d'}, \kappa)$-good. Since $\gcd(J') \nmid d$ we have $d' = \mathrm{lcm}(\gcd(J'), d) \geqslant 2d$. This implies the result. □

We are ready now to prove the final result

Lemma 22. *For $n \geqslant k^{2.5}$ there is $k' \leqslant k$ such that $\Pi(n, k') = \Omega((\sqrt{k}/\log^2 k) \log n)$.*

Proof. Let M be the $1 \times n$ matrix $[111 \cdots 1]$. Then M is $\mathbb{N}_2$-good. By Lemma 21, there exist $k_1, k_2, \cdots, k_t \leqslant k$, $t = O(\log k)$, $q_t = 1 + \Pi(k_1, n) + \cdots + \Pi(k_t, n)$, $d' \geqslant 2^{t+1} > K$ and a $q_t \times n$ matrix M' that is $(\mathbb{N}_{d'}, \kappa)$-good. Since $d' > K$, M' is $([K], \kappa)$-good. By Lemma 17,

$$q_t = \Omega\left(K \log \frac{n - \kappa K^2}{K}\right) = \Omega\left(\frac{\sqrt{k}}{\log k} \log n\right).$$

Therefore, there exists $k' := k_i \leqslant k$ such that

$$\Pi(n, k') = \Omega\left(\frac{\sqrt{k}}{\log^2 k} \log n\right).$$

□

References

[1] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Trans. Information Theory*, 51(11):4032–4039, 2005. URL: https://doi.org/10.1109/TIT.2005.856958, doi:10.1109/TIT.2005.856958.

[2] Roksana Baleshzar, Meiram Murzabulatov, Ramesh Krishnan S. Pallavoor, and Sofya Raskhodnikova. Testing unateness of real-valued functions. *CoRR*, abs/1608.07652, 2016. URL: http://arxiv.org/abs/1608.07652, arXiv:1608.07652.

[3] Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1021–1032, 2016. URL: https://doi.org/10.1145/2897518.2897567, doi:10.1145/2897518.2897567.

[4] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *Property Testing - Current Research and Surveys*, pages 269–275. 2010. URL: https://doi.org/10.1007/978-3-642-16367-8_19, doi:10.1007/978-3-642-16367-8_19.

[5] Eric Blais. Improved bounds for testing juntas. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Boston, MA, USA, August 25-27, 2008. Proceedings*, pages 317–330, 2008. URL: https://doi.org/10.1007/978-3-540-85363-3_26, doi:10.1007/978-3-540-85363-3_26.

[6] Eric Blais. Testing juntas nearly optimally. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 151–158, 2009. URL: https://doi.org/10.1145/1536414.1536437, doi:10.1145/1536414.1536437.

[7] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 210–220, 2011. URL: https://doi.org/10.1109/CCC.2011.31, doi:10.1109/CCC.2011.31.

[8] Eric Blais and Daniel M. Kane. Tight bounds for testing k-linearity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 435–446, 2012. URL: https://doi.org/10.1007/978-3-642-32512-0_37, doi:10.1007/978-3-642-32512-0_37.

[9] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. URL: https://doi.org/10.1016/0022-0000(93)90044-W, doi:10.1016/0022-0000(93)90044-W.

[10] Nader H. Bshouty. Almost optimal distribution-free junta testing. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, pages 2:1–2:13, 2019. URL: https://doi.org/10.4230/LIPIcs.CCC.2019.2, doi:10.4230/LIPIcs.CCC.2019.2.

[11] Harry Buhrman, David García-Soriano, Arie Matsliah, and Ronald de Wolf. The non-adaptive query complexity of testing k-parities. *Chicago J. Theor. Comput. Sci.*, 2013, 2013. URL: http://cjtcs.cs.uchicago.edu/articles/2013/6/contents.html.

[12] Deeparnab Chakrabarty and C. Seshadhri. A $o(n)$ monotonicity tester for boolean functions over the hypercube. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 411–418, 2013. URL: https://doi.org/10.1145/2488608.2488660, doi:10.1145/2488608.2488660.

[13] Deeparnab Chakrabarty and C. Seshadhri. A $\tilde{O}(n)$ non-adaptive tester for unateness. *CoRR*, abs/1608.06980, 2016. URL: http://arxiv.org/abs/1608.06980, arXiv:1608.06980.

[14] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, pages 545–556, 2011. URL: https://doi.org/10.1007/978-3-642-22006-7_46, doi:10.1007/978-3-642-22006-7_46.

[15] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 519–528, 2015. URL: https://doi.org/10.1145/2746539.2746570, doi:10.1145/2746539.2746570.

[16] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. *CoRR*, abs/1412.5655, 2014. URL: http://arxiv.org/abs/1412.5655, arXiv:1412.5655.

[17] Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond talagrand functions: new lower bounds for testing monotonicity and unateness. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 523–536, 2017. URL: https://doi.org/10.1145/3055399.3055461, doi:10.1145/3055399.3055461.

[18] Xi Chen, Erik Waingarten, and Jinyu Xie. Boolean unateness testing with $\widetilde{O}(n^{3/4})$ adaptive queries. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 868–879, 2017. URL: https://doi.org/10.1109/FOCS.2017.85, doi:10.1109/FOCS.2017.85.

[19] Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 549–558, 2007. URL: https://doi.org/10.1109/FOCS.2007.32, doi:10.1109/FOCS.2007.32.

[20] Bakir Farhi. An identity involving the least common multiple of binomial coefficients and its application. *The American Mathematical Monthly*, 116(9):836–839, 2009. URL: http://www.jstor.org/stable/40391302.

[21] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 103–112, 2002. URL: https://doi.org/10.1109/SFCS.2002.1181887, doi:10.1109/SFCS.2002.1181887.

[22] Oded Goldreich. On testing computability by small width obdds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, pages 574–587, 2010. URL: https://doi.org/10.1007/978-3-642-15369-3_43, doi:10.1007/978-3-642-15369-3_43.

[23] Oded Goldreich, editor. *Property Testing - Current Research and Surveys*, volume 6390 of *Lecture Notes in Computer Science*. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-16367-8, doi:10.1007/978-3-642-16367-8.

[24] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. URL: http://www.cambridge.org/us/catalogue/catalogue.asp?isbn=9781107194052, doi:10.1017/9781108135252.

[25] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. URL: https://doi.org/10.1007/s004930070011, doi:10.1007/s004930070011.

[26] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. URL: https://doi.org/10.1145/285055.285060, doi:10.1145/285055.285060.

[27] Parikshit Gopalan, Ryan O'Donnell, Rocco A. Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. *SIAM J. Comput.*, 40(4):1075–1100, 2011. URL: https://doi.org/10.1137/100785429, doi:10.1137/100785429.

[28] Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. URL: https://doi.org/10.1137/050645804, doi:10.1137/050645804.

[29] Thomas Hofmeister. An application of codes to attribute-efficient learning. In *Computational Learning Theory, 4th European Conference, EuroCOLT '99, Nordkirchen, Germany, March 29-31, 1999, Proceedings*, pages 101–110, 1999. URL: https://doi.org/10.1007/3-540-49097-3_9, doi:10.1007/3-540-49097-3_9.

[30] Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric type theorems. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 52–58, 2015. URL: https://doi.org/10.1109/FOCS.2015.13, doi:10.1109/FOCS.2015.13.

[31] Subhash Khot and Igor Shinkar. An $\widetilde{O}(n)$ queries adaptive tester for unateness. *CoRR*, abs/1608.02451, 2016. URL: http://arxiv.org/abs/1608.02451, arXiv:1608.02451.

[32] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing ±1-weight halfspace. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, pages 646–657, 2009. URL: https://doi.org/10.1007/978-3-642-03685-9_48, doi:10.1007/978-3-642-03685-9_48.

[33] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing halfspaces. *SIAM J. Comput.*, 39(5):2004–2047, 2010. URL: https://doi.org/10.1137/070707890, doi:10.1137/070707890.

[34] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM J. Discrete Math.*, 16(1):20–46, 2002. URL: http://epubs.siam.org/sam-bin/dbq/article/40744.

[35] Dana Ron. Property testing: A learning theory perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008. URL: https://doi.org/10.1561/2200000004, doi:10.1561/2200000004.

[36] Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009. URL: https://doi.org/10.1561/0400000029, doi:10.1561/0400000029.

[37] Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. *SIAM J. Discrete Math.*, 25(4):1562–1588, 2011. URL: https://doi.org/10.1137/100791075, doi:10.1137/100791075.

[38] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. URL: https://doi.org/10.1137/S0097539793255151, doi:10.1137/S0097539793255151.

[39] Mert Saglam. Near log-convexity of measured heat in (discrete) time and consequences. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 967–978, 2018. URL: https://doi.org/10.1109/FOCS.2018.00095, doi:10.1109/FOCS.2018.00095.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il