Almost Optimal Distribution-Free Junta Testing

Nader H. Bshouty
Department of Computer Science, Technion, Haifa, Israel
bshouty@cs.technion.ac.il

Abstract

We consider the problem of testing whether an unknown n-variable Boolean function is a k-junta in the distribution-free property testing model, where the distance between functions is measured with respect to an arbitrary and unknown probability distribution over $\{0,1\}^n$. Chen, Liu, Servedio, Sheng and Xie [35] showed that the distribution-free k-junta testing can be performed, with one-sided error, by an adaptive algorithm that makes $\tilde{O}(k^2)/\epsilon$ queries. In this paper, we give a simple two-sided error adaptive algorithm that makes $\tilde{O}(k/\epsilon)$ queries.

2012 ACM Subject Classification Mathematics of computing; Mathematics of computing → Discrete mathematics; Mathematics of computing → Probabilistic algorithms; Theory of computation → Probabilistic computation

Keywords and phrases Distribution-free property testing, k-Junta

Digital Object Identifier 10.4230/LIPIcs.CCC.2019.2

Related Version `https://arxiv.org/abs/1901.00717`

Acknowledgements We would like to thank Xi Chen for reading the early version of the paper and for verifying the correctness of the algorithm.

1 Inroduction

Property testing of Boolean function was first considered in the seminal works of Blum, Luby and Rubinfeld [11] and Rubinfeld and Sudan [42] and has recently become a very active research area. See for example, [1, 2, 3, 4, 7, 8, 13, 14, 15, 16, 18, 19, 22, 24, 27, 29, 32, 33, 37, 36, 39, 43] and other works referenced in the surveys [26, 40, 41].

A function $f : \{0,1\}^n \to \{0,1\}$ is said to be k-junta if it depends on at most k variables. Juntas have been of particular interest to the computational learning theory community [9, 10, 12, 30, 34, 38]. A problem closely related to learning juntas is the problem of testing juntas: Given black-box query access to a Boolean function f. Distinguish, with high probability, the case that f is k-junta versus the case that f is ϵ-far from every k-junta.

In the uniform distribution framework, where the distance between two functions is measured with respect to the uniform distribution, Ficher et al. [24] introduced the junta testing problem and gave adaptive and non-adaptive algorithms that make $poly(k)/\epsilon$ queries. Blais in [5] gave a non-adaptive algorithm that makes $\tilde{O}(k^{3/2})/\epsilon$ queries and in [6] an adaptive algorithm that makes $O(k \log k + k/\epsilon)$ queries. On the lower bounds side, Fisher et al. [24] gave an $\Omega(\sqrt{k})$ lower bound. Chockler and Gutfreund [21] gave an $\Omega(k)$ lower bound for adaptive testing and, recently, Sağlam in [43] improved this lower bound to $\Omega(k \log k)$. For the non-adaptive testing Chen et al. [17] gave the lower bound $\tilde{\Omega}(k^{3/2})/\epsilon$.

In the *distribution-free property testing*, [28], the distance between Boolean functions is measured with respect to an arbitrary and unknown distribution $\mathcal{D}$ over $\{0,1\}^n$. In this model, the testing algorithm is allowed (in addition to making black-box queries) to draw random $x \in \{0,1\}^n$ according to the distribution $\mathcal{D}$. This model is studied in [20, 23, 25, 31, 35]. For testing k-junta in this model, Chen et al. [35] gave a one-sided adaptive algorithm that makes $\tilde{O}(k^2)/\epsilon$ queries and proved a lower bound $\Omega(2^{k/3})$ for any non-adaptive algorithm.

34th Computational Complexity Conference (CCC 2019).
Editor: Amir Shpilka; Article No. 2; pp. 2:1–2:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

The results of Halevy and Kushilevitz [31] gives a one-sided non-adaptive algorithm that makes $O(2^k/\epsilon)$ queries. The adaptive $\Omega(k \log k)$ uniform-distribution lower bound from [43] trivially extend to the distribution-free model.

In this paper, we close the gap between the adaptive lower and upper bound. We prove

▶ **Theorem 1.** *For any $\epsilon > 0$, there is a two-sided distribution-free adaptive algorithm for ϵ-testing k-junta that makes $\tilde{O}(k/\epsilon)$ queries.*

Our exact upper bound is $O((k/\epsilon)\log(k/\epsilon))$ and therefore, by Sağlam [43] lower bound of $\Omega(k \log k)$, our bound is tight for any constant ϵ.

2 Preliminaries

In this section we give some notations follows by a formal definition of the model and some preliminary known results

2.1 Notations

We start with some notations. Denote $[n] = \{1, 2, \ldots, n\}$. For $S \subseteq [n]$ and $x = (x_1, \ldots, x_n)$ we write $x(S) = \{x_i | i \in S\}$. 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. When we write $x_I = 0$ we mean $x_I = 0_I$. For $X_1, X_2 \subseteq [n]$ where $X_1 \cap X_2 = \emptyset$ and $x \in \{0,1\}^{X_1}, y \in \{0,1\}^{X_2}$ we write $x \circ y$ to denote their concatenation, the string in $\{0,1\}^{X_1 \cup X_2}$ that agrees with x over coordinates in X_1 and agrees with y over X_2. For $X \subseteq [n]$ we denote $\overline{X} = [n] \backslash X$. We say that the Boolean function $f : \{0,1\}^n \to \{0,1\}$ is a literal if $f \in \{x_1, \ldots, x_n, \bar{x_1}, \ldots, \bar{x_n}\}$.

Given $f, g : \{0,1\}^n \to \{0,1\}$ and a probability distribution $\mathcal{D}$ over $\{0,1\}^n$, we say that f is *ϵ-close to g with respect to $\mathcal{D}$* if $\mathbf{Pr}_{x \in \mathcal{D}}[f(x) \neq g(x)] \leq \epsilon$, where $x \in \mathcal{D}$ means x is chosen from $\{0,1\}^n$ according to the distribution $\mathcal{D}$. We say that f is *ϵ-far from g with respect to $\mathcal{D}$* if $\mathbf{Pr}_{x \in \mathcal{D}}[f(x) \neq g(x)] \geq \epsilon$. We say that f is *ϵ-far from every k-junta with respect to $\mathcal{D}$* if for every k-junta g, f is ϵ-far from g with respect to $\mathcal{D}$. We will use U to denote the uniform distribution over $\{0,1\}^n$.

2.2 The Model

In this subsection, we define the model.

We consider the problem of testing juntas in the distribution-free testing model. In this model, the algorithm has access to a k-junta f via a black-box that returns $f(x)$ when a string x is queried, and access to unknown distribution $\mathcal{D}$ via an oracle that returns $x \in \{0,1\}^n$ chosen randomly according to the distribution $\mathcal{D}$.

A *distribution-free testing algorithm* $\mathcal{A}$ is a algorithm that, given as input a distance parameter ϵ and the above two oracles,

1. if f is k-junta then $\mathcal{A}$ output "accept" with probability at least $2/3$.
2. if f is ϵ-far from every k-junta with respect to the distribution $\mathcal{D}$ then it output "reject" with probability at least $2/3$.

We say that $\mathcal{A}$ is *one-sided* if it always accepts when f is k-junta, otherwise, it is called *two sided* algorithm. The *query complexity* of a distribution-free testing algorithm is the number of queries made on f.

2.3 Preliminaries Results

In this section, we give some known results that will be used in the sequel.

For a Boolean function f and $X \subset [n]$, we say that X is a *relevant set* of f if there are $a, b \in \{0,1\}^n$ such that $f(a) \neq f(b_X \circ a_{\overline{X}})$. When $X = \{i\}$ then we say that x_i is *relevant variable* of f. Obviously, if X is relevant set of f then $x(X)$ contains at least one relevant variable of f. In particular, we have

▶ **Lemma 2.** *If $\{X_i\}_{i\in[r]}$ is a partition of $[n]$ then for any Boolean function f the number of relevant sets X_i of f is at most the number of relevant variables of f.*

We will use the following folklore result that is formally proved in [35].

▶ **Lemma 3.** *Let $\{X_i\}_{i\in[r]}$ be a partition of $[n]$. Let f be a Boolean function and $u \in \{0,1\}^n$. If $f(u) \neq f(0)$ then a relevant set X_ℓ of f with a string $v \in \{0,1\}^n$ that satisfies $f(v) \neq f(0_{X_\ell} \circ v_{\overline{X_\ell}})$ can be found with $\lceil \log_2 r \rceil$ queries.*

The following is from [6]

▶ **Lemma 4.** *There exists a one-sided adaptive algorithm, **UniformJunta**(f, k, ϵ, δ), for ϵ-testing k-junta that makes $O(((k/\epsilon)+k\log k)\log(1/\delta))$ queries and rejects f with probability at least $1-\delta$ when it is ϵ-far from every k-junta with respect to the uniform distribution.*

The following is from [35].

▶ **Lemma 5.** *Let $\mathcal{D}$ be any probability distribution over $\{0,1\}^n$. If f is ϵ-far from every k-junta with respect to $\mathcal{D}$ then for any $J \subseteq [n]$, $|J| \leq k$ we have*

$$\boldsymbol{Pr}_{x\in\mathcal{D}, y\in U}[f(x) \neq f(x_J \circ y_{\bar{J}})] \geq \epsilon.$$

Proof. Let $J \subseteq [n]$ of size $|J| \leq k$. For every fixed $y \in \{0,1\}^n$ the function $f(x_J \circ y_{\bar{J}})$ is k-junta and therefore $\mathbf{Pr}_{x\in\mathcal{D}}[f(x) \neq f(x_J \circ y_{\bar{J}})] \geq \epsilon$. Therefore

$$\mathbf{Pr}_{x\in\mathcal{D}, y\in U}[f(x) \neq f(x_J \circ y_{\bar{J}})] \geq \epsilon.$$ ◀

3 The Algorithm

In this section, we prove the correctness of the algorithm and show that it makes $\tilde{O}(k/\epsilon)$ queries. We first give an overview of the algorithm then prove its correctness and analyze its query complexity.

3.1 Overview of the Algorithm

In this subsection we give an overview of the algorithm. We will use the notation in Subsection 2.1 and the definitions and Lemmas in Subsection 2.3.

Consider the algorithm in Figure 1. In steps 1-2, the algorithm uniformly at random partitions $[n]$ into $r = 2k^2$ disjoint sets $X_1, \ldots, X_r$. Lemma 6 shows that,

▶ **Fact 1.** *If the function is k-junta then with high probability (w.h.p), each set of variables $x(X_i) = \{x_j | j \in X_i\}$ contains at most one relevant variable.*

In steps 3-12, the algorithm finds

▶ **Fact 2.** *relevant sets $\{X_i\}_{i\in I}$ such that for $X = \cup_{i\in I} X_i$, w.h.p., the function $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-close to f with respect to $\mathcal{D}$.*

To find such set, the algorithm, after finding relevant sets $\{X_i\}_{i\in I'}$, chooses random string $u \in \mathcal{D}$ and tests if $f(u_{X'} \circ 0_{\overline{X'}}) \neq f(u)$ where $X' = \cup_{i\in I'} X_i$. The variable $t(X')$ counts for how many random strings $u \in \mathcal{D}$ we get $f(u_{X'} \circ 0_{\overline{X'}}) = f(u)$. If $t(X')$ reaches the value $O((\log k)/\epsilon)$ then, w.h.p, $f(x_{X'} \circ 0_{\overline{X'}})$ is $\epsilon/2$-close to f with respect to $\mathcal{D}$ and $X = X'$. Otherwise, $f(u_{X'} \circ 0_{\overline{X'}}) \neq f(u)$ and using Lemma 3 the algorithm finds a new relevant set X_ℓ. This is proved in Lemma 10.

In addition, for each relevant set X_ℓ, $\ell \in I$, it finds a string $v^{(\ell)}$ that satisfies $f(v^{(\ell)}) \neq f(0_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$. Obviously, if $|I| > k$ then, since each relevant set contains at least one relevant variable, the target is not k-junta and the algorithm rejects. See Lemma 2.

Now one of the key ideas is the following: If f is k-junta then $f(x_X \circ 0_{\overline{X}})$ is k-junta. If f is ϵ-far from every k-junta with respect to $\mathcal{D}$ then since, by Fact 2, w.h.p., $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-close to f with respect to $\mathcal{D}$ we have that,

▶ **Fact 3.** *If f is ϵ-far from every k-junta with respect to $\mathcal{D}$ then, w.h.p., $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-far from every k-junta with respect to $\mathcal{D}$.*

Now, since each X_ℓ, $\ell \in I$ is relevant set and $f(v^{(\ell)}) \neq f(0_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$, for $\ell \in I$ the function $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is non-constant. In steps 13-17, the algorithm tests that,

▶ **Fact 4.** *w.h.p., for each $\ell \in I$ there is $\tau(\ell) \in X_\ell$ such that $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is close to some literal in $\{x_{\tau(\ell)}, \overline{x_{\tau(\ell)}}\}$, with respect to the uniform distribution.*

This is done using the procedure **UniformJunta** in Lemma 4.

If f is k-junta then, by Fact 1 and 2, w.h.p., it passes this test (does not output reject). This is Lemma 7. If the algorithm does not pass this test, it rejects. If f is not k-junta and it passes this test, then the statement in Fact 4 is true. This is proved in Lemma 11.

Consider now steps 18-28. First, let us consider a function f that is ϵ-far from every k-junta with respect to $\mathcal{D}$. Let $J = \{\tau(\ell) \mid \ell \in I\}$ where $\tau(\ell)$ is as defined in Fact 4. Since by Fact 3, w.h.p., $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-far from every k-junta with respect to $\mathcal{D}$ and $|J| = |I| \leq k$, by Lemma 5, w.h.p.,

$$\mathbf{Pr}_{y\in U, x\in\mathcal{D}}[f(x_X \circ 0_{\overline{X}}) \neq f(x_J \circ y_{X\setminus J} \circ 0_{\overline{X}})] \geq \epsilon/2.$$

So we need to test whether $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-far from $f(x_J \circ y_{X\setminus J} \circ 0_{\overline{X}})$ (those are equal in the case when f is k-Junta). This is the last test we would like to do but the problem is that we do not know J, so we cannot use this test as is. So we change it, as is done in [35], to an equivalent test as follows

$$\mathbf{Pr}_{z\in U, x\in\mathcal{D}}[f(x_X \circ 0_{\overline{X}}) \neq f((x_X + z_X) \circ 0_{\overline{X}}) \mid z_J = 0_J] \geq \epsilon/2.$$

To be able to draw uniformly random z_X with $z_J = 0_J$, we use Fact 4, that is, the fact that each $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is close to one of the literals in $\{x_{\tau(\ell)}, \overline{x_{\tau(\ell)}}\}$. For every $\ell \in I$, the algorithm draws uniformly random $w := z_{X_\ell}$ and then using the fact that $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is close to one of the literals in $\{x_{\tau(\ell)}, \overline{x_{\tau(\ell)}}\}$ where $\tau(\ell) \in X_\ell$ the algorithm tests in which set $Y_{\ell,0} := \{j \in X_\ell \mid w_j = 0\}$ or $Y_{\ell,1} := \{j \in X_\ell \mid w_j = 1\}$ the index $\tau(\ell)$ falls. If $\tau(\ell) \in Y_{\ell,0}$ then the entry $\tau(\ell)$ in z_{X_ℓ} is zero and if $\tau(\ell) \in Y_{\ell,1}$ then the entry $\tau(\ell)$ in z_{X_ℓ} is one. In the latter case, the algorithm replaces z_{X_ℓ} with $\overline{z_{X_\ell}}$ (negation of each entry in z_{X_ℓ}) which is also uniformly random. This gives a random uniform z_{X_ℓ} with $z_{\tau(\ell)} = 0$. We do that for every $\ell \in I$ and get a random uniform z with $z_J = 0$. This is proved in Lemma 12. Then the algorithm rejects if $f(x_X \circ 0_{\overline{X}}) \neq f((x_X + z_X) \circ 0_{\overline{X}})$. If $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-far from every

k-junta then, by Lemma 5, $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-far from $f(x_J \circ y_{X\setminus J} \circ 0_{\overline{X}})$, and the algorithm, with one test, rejects with probability at least $\epsilon/2$. Therefore, by repeating this test $O(1/\epsilon)$ times the algorithm rejects w.h.p. This is proved in Lemma 13.

Now we consider f that is k-junta. Obviously, if f is k-junta then $f(x_X \circ 0_{\overline{X}}) = f((x_X + z_X) \circ 0_{\overline{X}})$ when $z_J = 0$ and the algorithm accepts. This is because $x(J)$ are the relevant variables in $f(x_X \circ 0_{\overline{X}})$. This is proved in Lemma 8.

3.2 The algorithm for k-Junta

In this subsection, we show that if the target function f is k-junta then the algorithm accepts with probability at least $2/3$.

We first prove

▶ **Lemma 6.** *Consider steps 1-2 in the algorithm. If f is a k-junta then, with probability at least $2/3$, for each $i \in [r]$, the set $x(X_i) = \{x_j | j \in X_i\}$ contains at most one relevant variable of f.*

Proof. Let x_{i_1} and x_{i_2} be two relevant variables in f. The probability that x_{i_1} and x_{i_2} are in the same set is equal to $1/r$. By the union bound, it follows that the probability that some relevant variables x_{i_1} and x_{i_2} in f are in the same set is at most $\binom{k}{2}/r \le 1/3$. ◀

We now show that w.h.p. the algorithm reaches the final test in the algorithm

▶ **Lemma 7.** *If f is k-junta and each $x(X_i)$ contains at most one relevant variable of f then*
1. *Each $x(X_i)$, $i \in I$, contains exactly one relevant variable.*
2. *The algorithm reaches step 18*

Proof. By Lemma 3 and steps 7-9, for $\ell \in I$, $f(v^{(\ell)}) \neq f(0_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ and therefore $x(X_\ell)$ contains exactly one relevant variable. Thus, for every $\ell \in I$, $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is a literal.

If the algorithm does not reach step 18, then it either halts in step 10, 15 or 17. If it halts in step 10 then $|I| > k$ and therefore, by Lemma 2, f contains more than k relevant variables and then it is not k-Junta. If it halts in step 15 then, by Lemma 4, for some X_ℓ, $\ell \in I$, $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is not 1-Junta (literal or constant function) and therefore X_ℓ contains at least two relevant variables. If it halts in step 17, then $f(b_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}) = f(\overline{b_{X_\ell}} \circ v^{(\ell)}_{\overline{X_\ell}})$ and then $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is not a literal. In all cases we get a contradiction. ◀

We now give two Lemmas that show that, with probability at least $2/3$, the algorithm accepts k-junta.

▶ **Lemma 8.** *If f is k-Junta and each $x(X_i)$ contains at most one relevant variable of f then the algorithm outputs "accept".*

Proof. By Lemma 7, the algorithm reaches step 18. We now show that it reaches step 29. Now we need to show that the algorithm does not halt in step 25 or 28.

Since $Y_{\ell,0}, Y_{\ell,1}$ is a partition of X_ℓ, $\ell \in I$ and X_ℓ contains exactly one relevant variable in $x(X_\ell)$ of f, this variable is either in $x(Y_{\ell,0})$ or in $x(Y_{\ell,1})$ but not in both. Suppose w.l.o.g. it is in $x(Y_{\ell,0})$ and not in $x(Y_{\ell,1})$. Then $f(x_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ is a literal and $f(x_{Y_{\ell,1}} \circ b_{Y_{\ell,0}} \circ v^{(\ell)}_{\overline{X_\ell}})$ is a constant function. This implies that for any b, $f(b_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}}) \neq f(\overline{b_{Y_{\ell,0}}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ and $f(b_{Y_{\ell,1}} \circ b_{Y_{\ell,0}} \circ v^{(\ell)}_{\overline{X_\ell}}) = f(\overline{b_{Y_{\ell,1}}} \circ b_{Y_{\ell,0}} \circ v^{(\ell)}_{\overline{X_\ell}})$. Therefore, $G_{\ell,0} = h$ and $G_{\ell,1} = 0$. Thus the algorithm does not halt in step 25.

Algorithm Simple$\mathcal{D}k-$Junta$(f, \mathcal{D}, \epsilon)$
Input: Oracle that accesses a Boolean function f and
oracle that draws a random $x \in \{0,1\}^n$ according to the distribution $\mathcal{D}$.
Output: Either "accept" or "reject"

Partition $[n]$ into r sets
1. Set $r = 2k^2$.
2. Choose uniformly at random a partition $X_1, X_2, \ldots, X_r$ of $[n]$

Find a close function and relevant sets
3. Set $X = \emptyset$; $I = \emptyset$; $t(X) = 0$
4. Repeat $M = 2k\ln(15k)/\epsilon$ times
5. Choose $u \in \mathcal{D}$.
6. $t(X) \leftarrow t(X) + 1$
7. If $f(u_X \circ 0_{\overline{X}}) \neq f(u)$ then
8. Binary search: find a new relevant set X_ℓ; $X \leftarrow X \cup X_\ell$; $I \leftarrow I \cup \{\ell\}$
9. and a string $v^{(\ell)} \in \{0,1\}^n$ such that $f(v^{(\ell)}) \neq f(0_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$.
10. If $|I| > k$ then output "reject" and halt.
11. $t(X) = 0$.
12. If $t(X) = 2\ln(15k)/\epsilon$ then Goto 13.

Tests if each relevant set is close to a literal
13. For every $\ell \in I$ do
14. If **UniformJunta**$(f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}), 1, 1/30, 1/15)$="reject"
15. then output "reject" and halt
16. Choose $b \in U$
17. If $f(b_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}) = f(\overline{b_{X_\ell}} \circ v^{(\ell)}_{\overline{X_\ell}})$ then output "reject" and halt

The final test of Lemma 5
18. Repeat $M' = (2\ln 15)/\epsilon$ times
19. Choose $w \in U$; $z = 0_{\overline{X}}$
20. For every $\ell \in I$ do
21. Set $Y_{\ell,\xi} = \{j \in X_\ell | w_j = \xi\}$ for $\xi \in \{0,1\}$.
22. Set $G_{\ell,0} = G_{\ell,1} = 0$;
23. Repeat $h = \ln(15M'k)/\ln(4/3)$ times
24. Choose $b \in U$;
If $f(b_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}}) \neq f(\overline{b_{Y_{\ell,0}}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ then $G_{\ell,0} \leftarrow G_{\ell,0} + 1$
If $f(b_{Y_{\ell,1}} \circ b_{Y_{\ell,0}} \circ v^{(\ell)}_{\overline{X_\ell}}) \neq f(\overline{b_{Y_{\ell,1}}} \circ b_{Y_{\ell,0}} \circ v^{(\ell)}_{\overline{X_\ell}})$ then $G_{\ell,1} \leftarrow G_{\ell,1} + 1$
25. If $(\{G_{\ell,0}, G_{\ell,1}\} \neq \{0, h\})$ then output "reject" and halt
26. If $G_{\ell,0} = h$ then $z \leftarrow z \circ w_{X_\ell}$ else $z \leftarrow z \circ \overline{w_{X_\ell}}$
27. Choose $u \in \mathcal{D}$
28. If $f(u_X \circ 0_{\overline{X}}) \neq f((u_X + z_X) \circ 0_{\overline{X}})$ then output "reject" and halt.
29. Output "accept"

Figure 1 A two-sided distribution-free adaptive algorithm for ϵ-testing k-junta.

Now for every X_ℓ, $\ell \in I$, let $\tau(\ell) \in X_\ell$ be such that $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}) \in \{x_{\tau(\ell)}, \overline{x_{\tau(\ell)}}\}$. If $\tau(\ell) \in Y_{\ell,0}$ then $G_{\ell,0} = h$ and then by step 26, $z_{\tau(\ell)} = w_{\tau(\ell)} = 0$. If $\tau(\ell) \in Y_{\ell,1}$ then $G_{\ell,1} = h$ and then $z_{\tau(\ell)} = \overline{w_{\tau(\ell)}} = 0$. Therefore for every relevant variable $x_{\tau(\ell)}$ in $\hat{f} = f(x_X \circ 0_{\overline{X}})$ we have $z_{\tau(\ell)} = 0$ which implies that $f(u_X \circ 0_{\overline{X}}) = f((u_X + z_X) \circ 0_{\overline{X}})$ and therefore the algorithm does not halt in step 28. ◀

▶ **Lemma 9.** *If f is k-Junta then the algorithm outputs "accept" with probability at least* $2/3$.

Proof. The result follows from Lemma 6 and Lemma 8. ◀

3.3 The Algorithm for ϵ-Far Functions

In this subsection, we prove that if f is ϵ-far from every k-junta then the algorithm rejects with probability at least $2/3$.

The first lemma shows that, w.h.p., $f(u_X \circ 0_{\overline{X}})$ is $\epsilon/2$-close to f.

▶ **Lemma 10.** *If the algorithm reaches step 13 then $t(X) = 2\ln(15k)/\epsilon$ and $|I| \le k$. If*

$$\boldsymbol{Pr}_{u\in\mathcal{D}}[f(u_X \circ 0_{\overline{X}}) \neq f(u)] \ge \epsilon/2$$

then the algorithm reaches step 13 with probability at most $1/15$.

Proof. The algorithm does not reaches step 13 if and only if it halts in step 10 and then $|I| > k$. The size of I is increased by one each time the condition, $f(u_X \circ 0_{\overline{X}}) \neq f(u)$, in step 7, is true. Therefore, if the algorithm reaches step 13 then the condition in step 7 was true at most k times and $|I| \le k$. Then steps 8-11 are executed at most k times. Thus, $t()$ is updated to 0 at most k times. The loop 5-12 is repeated M times and $t()$ is updated to 0 at most k times and therefore there is X for which $t(X) = M/k = 2\ln(15k)/\epsilon$. This implies that when the algorithm reaches step 13, we have $t(X) = 2\ln(15k)/\epsilon$.

The probability that the algorithm reaches step 13 with $\mathbf{Pr}_{u\in\mathcal{D}}[f(u_X \circ 0_{\overline{X}}) \neq f(u)] > \epsilon/2$ is the probability that for one (of the at most k) X', $\mathbf{Pr}_{u\in\mathcal{D}}[f(u_{X'} \circ 0_{\overline{X'}}) \neq f(u)] > \epsilon/2$ and $t(X') = 2\ln(15k)/\epsilon$. By the union bound, this probability is less than

$$k\left(1 - \frac{\epsilon}{2}\right)^{2\ln(15k)/\epsilon} = \frac{1}{15}.$$ ◀

In the following lemma we show that, w.h.p, each $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is close to a literal.

▶ **Lemma 11.** *Consider steps 13-15. If for some $\ell \in I$, $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-far from every literal with respect to the uniform distribution then, with probability at least $1 - (2/15)$, the algorithm rejects.*

Proof. If $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-far from every literal with respect to the uniform distribution then it is either (case 1) $(1/30)$-far from every 1-Junta (literal or constant) or (case 2) $(1/30)$-far from every literal and $(1/30)$-close to 0-Junta. In case 1, by Lemma 4, with probability at least $1 - (1/15)$, **UniformJunta** $(f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}), 1, 1/30, 1/15) =$ "reject" and then the algorithm rejects. In case 2, if $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $1/30$-close to some 0-Junta then it is either $(1/30)$-close to 0 or $(1/30)$-close to 1. Suppose it is $(1/30)$-close to 0. Let b be a random uniform string generated in steps 16. Then $\overline{b}$ is random uniform and for $g(x) = f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ we have

$$\begin{aligned}
\mathbf{Pr}[\text{The algorithm does not reject}] &= \mathbf{Pr}\left[g(b) \neq g(\overline{b})\right] \\
&= \mathbf{Pr}[g(b) = 1 \wedge g(\overline{b}) = 0] + \mathbf{Pr}[g(b) = 0 \wedge g(\overline{b}) = 1] \\
&\leq \mathbf{Pr}[g(b) = 1] + \mathbf{Pr}[g(\overline{b}) = 1] \\
&\leq \frac{1}{15}.
\end{aligned}$$

By the union bound the result follows. ◀

In the next lemma we prove that, w.h.p, the string z generated in steps 19-26 satisfies $z_J = 0$ where $x(J)$ are relevant variables of $f(u_X \circ 0_{\overline{X}})$.

▶ **Lemma 12.** *Consider steps 19-26. If for every $\ell \in I$ the function $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-close to a literal in $\{x_{\tau(\ell)}, \bar{x}_{\tau(\ell)}\}$ with respect to the uniform distribution, where $\tau(\ell) \in X_\ell$, and $\{G_{\ell,0}, G_{\ell,1}\} = \{0, h\}$ then, with probability at least $1 - k(3/4)^h$, we have: For every $\ell \in I$, $z_{\tau(\ell)} = 0$.*

Proof. Fix some ℓ. Suppose $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-close to $x_{\tau(\ell)}$ with respect to the uniform distribution. The case when it is $(1/30)$-close to $\overline{x_{\tau(\ell)}}$ is similar. Since $X_\ell = Y_{\ell,0} \cup Y_{\ell,1}$ and $Y_{\ell,0} \cap Y_{\ell,1} = \emptyset$ we have that $\tau(\ell) \in Y_{\ell,0}$ or $\tau(\ell) \in Y_{\ell,1}$, but not both. Suppose $\tau(\ell) \in Y_{\ell,0}$. The case where $\tau(\ell) \in Y_{\ell,1}$ is similar. Define the random variable $Z(x_{X_\ell}) = 1$ if $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}}) \neq x_{\tau(\ell)}$ and $Z(x_{X_\ell}) = 0$ otherwise. Then

$$\mathbf{E}_{x_{X_\ell} \in U}[Z(x_{X_\ell})] \leq \frac{1}{30}.$$

Therefore

$$\mathbf{E}_{x_{Y_{\ell,1}} \in U} \mathbf{E}_{x_{Y_{\ell,0}} \in U}[Z(x_{Y_{\ell,0}} \circ x_{Y_{\ell,1}})] \leq \frac{1}{30}$$

and by Markov's bound

$$\mathbf{Pr}_{x_{Y_{\ell,1}} \in U}\left[\mathbf{E}_{x_{Y_{\ell,0}} \in U}[Z(x_{Y_{\ell,0}} \circ x_{Y_{\ell,1}})] \geq \frac{2}{15}\right] \leq \frac{1}{4}.$$

That is, for a random uniform string $b \in \{0,1\}^n$, with probability at least $3/4$, $f(x_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(2/15)$-close to $x_{\tau(\ell)}$ with respect to the uniform distribution. Now, given that $f(x_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(2/15)$-close to $x_{\tau(\ell)}$ with respect to the uniform distribution the probability that $G_{\ell,0} = 0$ is the probability that $f(b_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}}) = f(\overline{b_{Y_{\ell,0}}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ for h random uniform strings $b \in \{0,1\}^n$. Let $b^{(1)}, \ldots, b^{(h)}$ be h random uniform strings in $\{0,1\}^n$, $V(b)$ be the event $f(b_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}}) = f(\overline{b_{Y_{\ell,0}}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ and A the event that $f(x_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(2/15)$-close to $x_{\tau(\ell)}$ with respect to the uniform distribution. Let $g(x_{Y_{\ell,0}}) = f(x_{Y_{\ell,0}} \circ b_{Y_{\ell,1}} \circ v^{(\ell)}_{\overline{X_\ell}})$. Then

$$\begin{aligned}
\mathbf{Pr}[V(b)|A] &= \mathbf{Pr}[g(b_{Y_{\ell,0}}) = g(\overline{b_{Y_{\ell,0}}})|A] \\
&= \mathbf{Pr}[(g(b_{Y_{\ell,0}}) = b_{\tau(\ell)} \wedge g(\overline{b_{Y_{\ell,0}}}) = b_{\tau(\ell)}) \vee \\
&\qquad\qquad (g(b_{Y_{\ell,0}}) = \overline{b_{\tau(\ell)}} \wedge g(\overline{b_{Y_{\ell,0}}}) = \overline{b_{\tau(\ell)}})|A] \\
&\leq \mathbf{Pr}[g(\overline{b_{Y_{\ell,0}}}) \neq \overline{b_{\tau(\ell)}} \vee g(b_{Y_{\ell,0}}) \neq b_{\tau(\ell)})|A] \\
&\leq \mathbf{Pr}[g(\overline{b_{Y_{\ell,0}}}) \neq \overline{b_{\tau(\ell)}}|A] + \mathbf{Pr}[g(b_{Y_{\ell,0}}) \neq b_{\tau(\ell)})|A] \leq \frac{4}{15}.
\end{aligned}$$

Since $\tau(\ell) \in Y_{\ell,0}$, we have $w_{\tau(\ell)} = 0$. Therefore, by step 26 and since $\tau(\ell) \in X_\ell$,

$$\begin{aligned}
\mathbf{Pr}[z_{\tau(\ell)} = 1] &= \mathbf{Pr}[G_{\ell,0} = 0 \wedge G_{\ell,1} = h] \\
&\le \mathbf{Pr}[G_{\ell,0} = 0] = \mathbf{Pr}[(\forall j \in [h]) V(b^{(j)})] \\
&= (\mathbf{Pr}[V(b)])^h \le \left(\mathbf{Pr}[V(b)|A] + \mathbf{Pr}[\overline{A}]\right)^h \le (4/15 + 1/4)^h \le (3/4)^h
\end{aligned}$$

Therefore, the probability that $z_{\tau(\ell)} = 1$ for some $\ell \in I$ is at most $k(3/4)^h$. ◀

We now show that w.h.p the algorithm reject if f is ϵ-far from every k-junta

▶ **Lemma 13.** *If f is ϵ-far from every k-junta with respect to $\mathcal{D}$ then, with probability at least $2/3$, the algorithm outputs "reject".*

Proof. If the algorithm stops in step 10 then we are done. Therefore we may assume that

$$|I| \le k. \tag{1}$$

By Lemma 10, if $\mathbf{Pr}_{u\in\mathcal{D}}[f(u_X \circ 0_{\overline{X}}) \ne f(u)] \ge \epsilon/2$ then, with probability at most $1/15$, the algorithm reaches step 13. So we may assume that (failure probability $1/15$)

$$\mathbf{Pr}_{u\in\mathcal{D}}[f(u_X \circ 0_{\overline{X}}) \ne f(u)] \le \epsilon/2. \tag{2}$$

Since f is ϵ-far from every k-junta with respect to $\mathcal{D}$ and $f(x_X \circ 0_{\overline{X}})$ is $\epsilon/2$-close to f with respect to $\mathcal{D}$ we have $f(x_X \circ 0_{\overline{X}})$ is $(\epsilon/2)$-far from every k-junta with respect to $\mathcal{D}$. Therefore, by Lemma 5,

$$\mathbf{Pr}_{u\in\mathcal{D}, y\in U}[f(u_X \circ 0_{\overline{X}}) = f(u_I \circ y_{X\setminus I} \circ 0_{\overline{X}})] \ge 1 - \frac{\epsilon}{2}. \tag{3}$$

By Lemma 11, if some $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-far from any literal with respect to the uniform distribution then, with probability at least $1 - (2/15)$, the algorithm rejects. So we may assume (failure probability $2/15$) that every $f(x_{X_\ell} \circ v^{(\ell)}_{\overline{X_\ell}})$ is $(1/30)$-close to some $x_{\tau(\ell)}$ or $\overline{x_{\tau(\ell)}}$ with respect to the uniform distribution, where $\tau(\ell) \in X_\ell$.

Let $z^{(1)}, \ldots, z^{(M')}$ be the strings generated in step 26. By Lemma 12, with probability at least $1 - M'k(3/4)^h \ge 1 - (1/15)$, every $z^{(i)}$ generated in step 26 satisfies $z^{(i)}_{\tau(\ell)} = 0$ for all $\ell \in I$. Also, since the distribution of w_{X_ℓ} and $\overline{w_{X_\ell}}$ is uniform, the distribution of $z^{(i)}_{X\setminus I}$ and $u_{X\setminus I} + z^{(i)}_{X\setminus I}$ is uniform. We now assume (failure probability $1/15$) that $z^{(i)}_I = 0$ for all i. Therefore, by (3),

$$\begin{aligned}
\mathbf{Pr}_{u\in\mathcal{D}, z^{(i)}_{X\setminus I}\in U}[(\forall i) f(u_X \circ 0_{\overline{X}})] &= f((u_X + z^{(i)}_X) \circ 0_{\overline{X}})] \\
&= \left(\mathbf{Pr}_{u\in\mathcal{D}, z^{(1)}_{X\setminus I}\in U}[f(u_X \circ 0_{\overline{X}}) = f((u_X + z^{(1)}_X) \circ 0_{\overline{X}})]\right)^{M'} \\
&= \left(\mathbf{Pr}_{u\in\mathcal{D}, y\in U}[f(u_X \circ 0_{\overline{X}}) = f(u_I \circ y_{X\setminus I} \circ 0_{\overline{X}})]\right)^{M'} \\
&\le (1 - \epsilon/2)^{M'} \le \frac{1}{15}.
\end{aligned}$$

Therefore, the failure probability of an output "reject" is at most $1/15 + 2/15 + 1/15 + 1/15 = 1/3$. ◀

3.4 The Query Complexity of the Algorithm

In this section we show that

▶ **Lemma 14.** *The query complexity of the algorithm is*

$$\tilde{O}\left(\frac{k}{\epsilon}\right).$$

Proof. The condition in step 7 requires two queries and is executed at most $M = 2k\ln(15k)/\epsilon$ times. This is $2M = O((k\log k)/\epsilon)$ queries. Steps 8 is executed at most $k+1$ times. This is because each time it is executed, the value of $|I|$ is increased by one, and when $|I| = k+1$ the algorithm rejects. By Lemma 3, to find a new relevant set the algorithm makes $O(\log r) = O(\log k)$ queries. This is $O(k\log k)$ queries. Steps 14 and 17 are executed $|I| \leq k$ times, and by Lemma 4, the total number of queries made is $O(1/(1/30)\log(15))k+2k = O(k)$.

The final test in the algorithm is repeated $M' = (2\ln 15)/\epsilon$ times (step 18) and each time, and for each $\ell \in I$, (step 20) it repeats h times (step 23) two conditions that takes 2 queries each (step 24). This takes $4M'kh = O((k/\epsilon)\ln(k/\epsilon))$ queries. The number of queries in step 28 is $2M' = O(1/\epsilon)$. Therefore the total number of queries is

$$O\left(\frac{k}{\epsilon}\ln\frac{k}{\epsilon}\right).$$ ◀

4 Open Problems

In this paper we proved that for any $\epsilon > 0$, there is a two-sided distribution-free adaptive algorithm for ϵ-testing k-junta that makes $\tilde{O}(k/\epsilon)$ queries. It is also interesting to find a one-sided distribution-free adaptive algorithm with such query complexity.

Chen et al. [35] proved the lower bound $\Omega(2^{k/3})$ for any non-adaptive (one round) algorithm. What is the minimal number rounds one needs to get $poly(k/\epsilon)$ query complexity? Can $O(1)$-round algorithms solve the problem with $poly(k/\epsilon)$ queries?

In the uniform distribution framework, where the distance between two functions is measured with respect to the uniform distribution Blais in [5] gave a non-adaptive algorithm that makes $\tilde{O}(k^{3/2})/\epsilon$ queries and in [6] an adaptive algorithm that makes $O(k\log k + k/\epsilon)$ queries. On the lower bounds side, Sağlam in [43] gave an $\Omega(k\log k)$ lower bound for adaptive testing and Chen et al. [17] gave an $\tilde{\Omega}(k^{3/2})/\epsilon$ lower bound for the non-adaptive testing. Thus in both the adaptive and non-adaptive uniform distribution settings, the query complexity of k-junta testing has now been pinned down to within logarithmic factors. It is interesting to study $O(1)$-round algorithms. For example, what is the query complexity for 2-round algorithm.

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. 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. 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. 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. Springer, 2010. 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. 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. 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. 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. doi:10.1007/978-3-642-32512-0_37.

9 Avrim Blum. Learning a Function of r Relevant Variables. In *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, pages 731–733, 2003. doi:10.1007/978-3-540-45167-9_54.

10 Avrim Blum and Pat Langley. Selection of Relevant Features and Examples in Machine Learning. *Artif. Intell.*, 97(1-2):245–271, 1997. doi:10.1016/S0004-3702(97)00063-5.

11 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. doi:10.1016/0022-0000(93)90044-W.

12 Nader H. Bshouty and Areej Costa. Exact learning of juntas from membership queries. *Theor. Comput. Sci.*, 742:82–97, 2018. doi:10.1016/j.tcs.2017.12.032.

13 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. doi:10.1145/2488608.2488660.

14 Deeparnab Chakrabarty and C. Seshadhri. A $\tilde{O}(n)$ Non-Adaptive Tester for Unateness. *CoRR*, abs/1608.06980, 2016. arXiv:1608.06980.

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. 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. arXiv:1412.5655.

17 Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the Query Complexity of Non-Adaptive Junta Testing. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 26:1–26:19, 2017. doi:10.4230/LIPIcs.CCC.2017.26.

18 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. doi:10.1145/3055399.3055461.

19 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. doi:10.1109/FOCS.2017.85.

20 Xi Chen and Jinyu Xie. Tight Bounds for the Distribution-Free Testing of Monotone Conjunctions. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 54–71, 2016. doi:10.1137/1.9781611974331.ch5.

21 Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Inf. Process. Lett.*, 90(6):301–305, 2004. doi:10.1016/j.ipl.2004.01.023.

22 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. doi:10.1109/FOCS.2007.32.

23 Elya Dolev and Dana Ron. Distribution-Free Testing for Monomials with a Sublinear Number of Queries. *Theory of Computing*, 7(1):155–176, 2011. doi:10.4086/toc.2011.v007a011.

24 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002. doi:10.1145/509907.509977.

25 Dana Glasner and Rocco A. Servedio. Distribution-Free Testing Lower Bound for Basic Boolean Functions. *Theory of Computing*, 5(1):191–216, 2009. doi:10.4086/toc.2009.v005a010.

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

27 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing Monotonicity. *Combinatorica*, 20(3):301–337, 2000. doi:10.1007/s004930070011.

28 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property Testing and its Connection to Learning and Approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

29 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. doi:10.1137/100785429.

30 David Guijarro, Jun Tarui, and Tatsuie Tsukiji. Finding Relevant Variables in PAC Model with Membership Queries. In *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 6-8, 1999, Proceedings*, page 313, 1999. doi:10.1007/3-540-46769-6_26.

31 Shirley Halevy and Eyal Kushilevitz. Distribution-Free Property-Testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. doi:10.1137/050645804.

32 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. doi:10.1109/FOCS.2015.13.

33 Subhash Khot and Igor Shinkar. An $\widetilde{O}(n)$ Queries Adaptive Tester for Unateness. *CoRR*, abs/1608.02451, 2016. arXiv:1608.02451.

34 Richard J. Lipton, Evangelos Markakis, Aranyak Mehta, and Nisheeth K. Vishnoi. On the Fourier Spectrum of Symmetric Boolean Functions with Applications to Learning Symmetric Juntas. In *20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA*, pages 112–119, 2005. doi:10.1109/CCC.2005.19.

35 Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 749–759, 2018. doi:10.1145/3188745.3188842.

36 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. doi:10.1007/978-3-642-03685-9_48.

37 Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing Halfspaces. *SIAM J. Comput.*, 39(5):2004–2047, 2010. doi:10.1137/070707890.

38 Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning functions of k relevant variables. *J. Comput. Syst. Sci.*, 69(3):421–434, 2004. doi:10.1016/j.jcss.2004.04.002.

39 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.

40 Dana Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008. doi:10.1561/2200000004.

41 Dana Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009. doi:10.1561/0400000029.

42 Ronitt Rubinfeld and Madhu Sudan. Robust Characterizations of Polynomials with Applications to Program Testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

43 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. doi:10.1109/FOCS.2018.00095.