

# Improved Bounds for Testing Juntas

Eric Blais\*

Carnegie Mellon University  
eblais@cs.cmu.edu

**Abstract.** We consider the problem of testing functions for the property of being a  $k$ -junta (i.e., of depending on at most  $k$  variables). Fischer, Kindler, Ron, Safra, and Samorodnitsky (*J. Comput. Sys. Sci.*, 2004) showed that  $\tilde{O}(k^2)/\epsilon$  queries are sufficient to test  $k$ -juntas, and conjectured that this bound is optimal for non-adaptive testing algorithms. Our main result is a non-adaptive algorithm for testing  $k$ -juntas with  $\tilde{O}(k^{3/2})/\epsilon$  queries. This algorithm disproves the conjecture of Fischer et al.

We also show that the query complexity of non-adaptive algorithms for testing juntas has a lower bound of  $\min(\tilde{\Omega}(k/\epsilon), 2^k/k)$ , essentially improving on the previous best lower bound of  $\Omega(k)$ .

## 1 Introduction

A function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is said to be a  $k$ -junta if it depends on at most  $k$  variables. Juntas provide a clean model for studying learning problems in the presence of many irrelevant features [4, 6], and have consequently been of particular interest to the computational learning theory community [5, 6, 12, 16, 17]. A problem closely related to learning juntas is the problem of *testing juntas*: given query access to a function, is it possible to efficiently determine if all but at most  $k$  of the variables in the function represent irrelevant features?

We consider the problem of testing juntas in the standard framework of property testing, as originally introduced by Rubinfeld and Sudan [19]. In this framework, we say that a function  $f$  is  $\epsilon$ -far from being a  $k$ -junta if for every  $k$ -junta  $g$ , the functions  $f$  and  $g$  disagree on at least an  $\epsilon$  fraction of inputs. A randomized algorithm  $\mathcal{A}$  that makes  $q$  queries to its input function is an  $\epsilon$ -testing algorithm for  $k$ -juntas if

1. All  $k$ -juntas are accepted by  $\mathcal{A}$  with probability at least  $2/3$ , and
2. All functions that are  $\epsilon$ -far from being  $k$ -juntas are rejected by  $\mathcal{A}$  with probability at least  $2/3$ .

A testing algorithm  $\mathcal{A}$  is *non-adaptive* if it does not use the answers of some queries to determine later queries; otherwise, the algorithm  $\mathcal{A}$  is *adaptive*.

In this article we consider the problem of determining the query complexity for the problem of testing juntas: given fixed  $k \geq 1$  and  $\epsilon > 0$ , what is the

---

\* Supported in part by a scholarship from the Fonds québécois de recherche sur la nature et les technologies (FQRNT).

minimum number  $q = q(k, \epsilon)$  of queries required for any algorithm  $\mathcal{A}$  to  $\epsilon$ -test  $k$ -juntas?

**Background.** The first result related to testing juntas was obtained by Bellare, Goldreich, and Sudan [2] in the context of testing long codes. That result was generalized by Parnas, Ron, and Samorodnitsky [18] to obtain an algorithm for  $\epsilon$ -testing 1-juntas with only  $O(1/\epsilon)$  queries.

The next important step in testing  $k$ -juntas was taken by Fischer, Kindler, Ron, Safra, and Samorodnitsky [10], who developed multiple algorithms for testing  $k$ -juntas with  $\text{poly}(k)/\epsilon$  queries. Those algorithms were particularly significant for showing explicitly that testing juntas can be done with a query complexity *independent* of the total number of variables. The most query-efficient algorithms they presented require  $\tilde{O}(k^2)/\epsilon$  queries<sup>1</sup> to  $\epsilon$ -test  $k$ -juntas.

Fischer et al. [10] also gave the first non-trivial lower bound on the query complexity for the testing juntas problem. They showed that any non-adaptive algorithm for  $\epsilon$ -testing  $k$ -juntas requires at least  $\tilde{\Omega}(\sqrt{k})$  queries and conjectured that the true query complexity for non-adaptive algorithms is  $k^2/\epsilon$  queries.

Chockler and Gutfreund [8] improved the lower bound for testing juntas by showing that all algorithms – adaptive or non-adaptive – for  $\epsilon$ -testing  $k$ -juntas require  $\Omega(k)$  queries. This result applies for all values of  $\epsilon < 1/8$ , but the bound itself does not increase as  $\epsilon$  decreases.

**Our results and techniques.** Our main result is an improvement on the upper bound for the query complexity of the junta testing problem.

**Theorem 1.1.** *The property of being a  $k$ -junta can be  $\epsilon$ -tested by a non-adaptive algorithm with  $\tilde{O}(k^{3/2})/\epsilon$  queries.*

The new algorithm presented in this article is the first for testing juntas with a number of queries sub-quadratic in  $k$ , and disproves the lower bound conjecture of Fischer et al.

Our algorithm is based on an algorithm of Fischer et al. for testing juntas [10, §4.2]). The observation that led to the development of the new algorithm is that the algorithm of Fischer et al. can be broken up into two separate tests: a “block test” and a simple “sampling test”. In this article, we generalize the sampling test, and we establish a structural Lemma for functions that are  $\epsilon$ -far from being  $k$ -juntas to show how the two tests can be combined to  $\epsilon$ -test  $k$ -juntas more efficiently.

Our second result is an improved lower bound on the number of queries required for testing juntas with non-adaptive algorithms. The new bound is the first lower bound for the query complexity of the junta testing problem that incorporates the accuracy parameter  $\epsilon$ .

---

<sup>1</sup> Here and in the rest of this article, the  $\tilde{O}(\cdot)$  notation is used to hide polylog factors. (i.e.,  $\tilde{O}(f(x)) = O(f(x) \log^c f(x))$  and  $\tilde{\Omega}(f(x)) = \Omega(\frac{f(x)}{\log^c f(x)})$  for some  $c \geq 0$ .)

**Theorem 1.2.** *Any non-adaptive algorithm for  $\epsilon$ -testing  $k$ -juntas must make at least  $\min\left(\Omega\left(\frac{k/\epsilon}{\log k/\epsilon}\right), \Omega\left(\frac{2^k}{k}\right)\right)$  queries.*

We prove Theorem 1.2 via Yao’s Minimax Principle [20]. The proof involves an extension of the argument of Chockler and Gutfreund [8] and an application of the Edge-Isoperimetric Inequality of Harper [13], Bernstein [3], and Hart [14].

**Organization.** We introduce some notation and definitions in Section 2. We present the new algorithm for  $\epsilon$ -testing  $k$ -juntas and its analysis in Section 3. In Section 4, we present the proof for the lower bound on the query complexity of non-adaptive algorithms for testing juntas. Finally, we conclude with some remarks and open problems in Section 5.

## 2 Preliminaries

**Notation.** For  $n \geq 1$ , let  $[n] = \{1, \dots, n\}$ . For a set  $A \subseteq [n]$ , we write  $\bar{A} = [n] \setminus A$  to represent the complement of  $A$  in  $[n]$ . When  $x, y \in \{0, 1\}^n$ , we define  $x_A y_{\bar{A}}$  to be the hybrid string  $z$  where  $z_i = x_i$  for every  $i \in A$  and  $z_j = y_j$  for every  $j \in \bar{A}$ .

We write  $\Pr_x[\cdot]$  (resp.,  $\mathbf{E}_x[\cdot]$ ) to denote the probability (resp., expectation) over the choice of  $x$  taken uniformly at random from  $\{0, 1\}^n$ . We also write  $H_k = \sum_{j=1}^k \frac{1}{j}$  to denote the  $k$ -th harmonic number.

**Variation.** In the analysis of the new algorithm for testing juntas, we consider the *variation* of sets of coordinates in a function, a concept introduced by Fischer et al. [10].<sup>2</sup>

**Definition 2.1.** *The variation of the set  $S \subseteq [n]$  of coordinates in the function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is*

$$\text{Vr}_f(A) = \Pr_x[f(x) \neq f(x_{\bar{S}}y_S)].$$

We write  $\text{Vr}_f(i) = \text{Vr}_f(\{i\})$  to represent the variation of the  $i$ th coordinate. The variation of a single coordinate is equivalent to the notion of *influence*, as defined in, e.g., [15].

Some useful properties of variation are its monotonicity, subadditivity, and submodularity.

**Fact 2.2 (Fischer et al. [10])** *For any function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , and any sets  $A, B, C \subseteq [n]$ , the following three properties hold:*

- (i) *Monotonicity:*  $\text{Vr}_f(A) \leq \text{Vr}_f(A \cup B)$
- (ii) *Subadditivity:*  $\text{Vr}_f(A \cup B) \leq \text{Vr}_f(A) + \text{Vr}_f(B)$
- (iii) *Submodularity:*  $\text{Vr}_f(A \cup B) - \text{Vr}_f(B) \geq \text{Vr}_f(A \cup B \cup C) - \text{Vr}_f(B \cup C)$

<sup>2</sup> The definition of variation used in [10] is slightly different, but is equivalent to the one used in this article up to a constant factor.

**The Independence Test.** A function  $f$  is said to be *independent* of a set  $S \subseteq [n]$  of coordinates if  $\text{Vr}_f(S) = 0$ . The definition of variation suggests a natural test for independence:

INDEPENDENCETEST [10]: Given a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and a set  $S \subseteq [n]$ , generate two inputs  $x, y \in \{0, 1\}^n$  independently and uniformly at random. If  $f(x) = f(x_{\bar{S}}y_S)$ , then accept; otherwise, reject.

Let us define  $\text{INDEPENDENCETEST}(f, S, m)$  to be the algorithm that runs  $m$  instances of the INDEPENDENCETEST on  $f$  and  $S$  and accepts if and only if every instance of the INDEPENDENCETEST accepts. By the definition of variation, this algorithm accepts with probability  $(1 - \text{Vr}_f(S))^m$ . In particular, this test always accepts when  $f$  is independent of the set  $S$  of coordinates, and rejects with probability at least  $1 - \delta$  when  $\text{Vr}_f(S) \geq \ln(1/\delta)/m$ .

### 3 The Algorithm for Testing Juntas

In this section, we present the algorithm for  $\epsilon$ -testing  $k$ -juntas with  $\tilde{O}(k^{3/2})/\epsilon$  queries. The algorithm has two main components: the BLOCKTEST and the SAMPLINGTEST. We introduce the BLOCKTEST in Section 3.1 and the SAMPLINGTEST in Section 3.2. Finally, in Section 3.3 we show how to combine both tests to obtain an algorithm for testing juntas.

#### 3.1 The Block Test

The purpose of the BLOCKTEST is to accept  $k$ -juntas and reject functions that have at least  $k + 1$  coordinates with “large” variation.

The BLOCKTEST first randomly partitions the coordinates in  $[n]$  into  $s$  sets  $I_1, \dots, I_s$ . It then applies the INDEPENDENCETEST to blocks of these sets to identify the sets of coordinates that have low variation. The test accepts if all but at most  $k$  of the sets  $I_1, \dots, I_s$  are identified as having low variation. The full algorithm is presented in Fig. 1.

The BLOCKTEST is based on Fischer et al.’s non-adaptive algorithm for testing juntas [10, §4.2], which uses a very similar test.<sup>3</sup> As the following two Propositions show, with high probability the BLOCKTEST accepts  $k$ -juntas and rejects functions with  $k + 1$  coordinates with variation at least  $\eta$ .

**Proposition 3.1 (Completeness).** *Fix  $\eta > 0$ , and let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a  $k$ -junta. Then the BLOCKTEST accepts  $f$  with probability at least  $1 - \delta$ .*

*Proof.* Let  $I_j$  be a set that contains only coordinates  $i$  with variation  $\text{Vr}_f(i) = 0$ . In a given round, the probability that  $I_j$  is included in  $B_T$  and none of the sets

<sup>3</sup> The principal difference between our version of the BLOCKTEST and Fischer et al.’s version of the test is that in [10], the set  $T$  is generating by including exactly  $k$  indices chosen at random from  $[s]$ .

BLOCKTEST( $f, k, \eta, \delta$ )  
*Additional parameters:*  $s = \lceil 2k^2/\delta \rceil$ ,  $r = \lceil 4k \ln(s/\delta) \rceil$ ,  $m = \lceil \ln(2r/\delta)/\eta \rceil$

1. Randomly partition the coordinates in  $[n]$  into  $s$  sets  $I_1, \dots, I_s$ .
2. For each of  $r$  rounds,
  - 2.1. Pick a random subset  $T \subseteq [s]$  by including each index independently with probability  $1/k$ .
  - 2.2. Define the block of coordinates  $B_T = \bigcup_{j \in T} I_j$ .
  - 2.3. If INDEPENDENCETEST( $f, B_T, m$ ) accepts, mark  $I_j$  as “variation-free” for every  $j \in T$ .
3. **Accept**  $f$  if at most  $k$  of the sets  $I_1, \dots, I_s$  are not marked as “variation-free”; otherwise **reject**  $f$ .

**Fig. 1.** The algorithm for the block test.

$I_j$  that contain a coordinate with positive variation are included in  $B_T$  is at least  $(1/k)(1 - 1/k)^k \geq 1/4k$  since  $(1 - 1/k)^k \geq 1/4$  for all  $k \geq 2$ . So the probability that  $I_j$  is not marked as “variation-free” in any of the  $r$  rounds is at most  $(1 - 1/4k)^r \leq e^{-r/4k} \leq \delta/s$  when  $r \geq 4k \ln(s/\delta)$ . By the union bound, all the sets  $I_j$  that contain only coordinates with no variation are identified as “variation-free” with probability at least  $1 - s(\delta/s) = 1 - \delta$ .  $\square$

**Proposition 3.2 (Soundness).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a function for which there exists a set  $S \subseteq [n]$  of size  $|S| = k + 1$  such that every coordinate  $i \in S$  has variation  $\text{Vr}_f(i) \geq \eta$ . Then the BLOCKTEST rejects  $f$  with probability at least  $1 - \delta$ .*

*Proof.* There are two ways in which the block test can wrongly accept the input function. The first way it can do so is by mapping all the coordinates with variation at least  $\eta$  into at most  $k$  sets during the random partition. We can upper bound the probability of this event with the probability that any collision occurs during the mapping of the first  $k + 1$  coordinates with high variation, which is at most  $\frac{1}{s} + \frac{2}{s} + \dots + \frac{k}{s} = \frac{k(k+1)}{2s} \leq \frac{k^2}{s} \leq \delta/2$ .

The second way in which the block test can wrongly accept the input function is by erroneously marking one of the sets  $I_j$  that contains a coordinate with variation at least  $\eta$  as “variation-free”. To bound the probability of this event happening, consider a given round in which  $B_T$  contains at least one of the coordinates  $i$  with variation  $\text{Vr}_f(i) \geq \eta$ . By Fact 2.2 (i), the variation of  $B_T$  is at least  $\eta$ , so when  $m \geq \ln(2r/\delta)/\eta$ , the INDEPENDENCETEST accepts  $B_T$  with probability at most  $\delta/2r$ . By the union bound, the probability that one of the  $r$  rounds results in a false “variation-free” marking is at most  $\delta/2$ . So the total probability that the algorithm wrongly accepts the function  $f$  is at most  $\delta/2 + \delta/2 = \delta$ .  $\square$

The BLOCKTEST algorithm makes  $2m$  queries to  $f$  in each round, so the total query complexity of the algorithm is  $2rm = O(k \log^2(k/\delta)/\eta)$ .

SAMPLINGTEST( $f, k, l, \eta, \delta$ )  
*Additional parameters:*  $r = \lceil 128k^2 \ln(2/\delta)/l^2 \rceil$ ,  $m = \lceil \ln(2r/\delta)/\eta \rceil$

1. Initialize the success counter  $c \leftarrow 0$ .
2. For each of  $r$  rounds,
  - 2.1. Pick a random subset  $T \subseteq [n]$  by including each coordinate independently with probability  $1/k$ .
  - 2.2. If INDEPENDENCETEST( $f, T, m$ ) accepts, set  $c \leftarrow c + 1$ .
3. **Accept**  $f$  if  $c/r \geq (1 - 1/k)^k - l/16k$ ; otherwise **reject**  $f$ .

**Fig. 2.** The algorithm for the sampling test.

### 3.2 The Sampling Test

The purpose of the SAMPLINGTEST is to accept  $k$ -juntas and reject functions that have a large number of coordinates with non-zero variation.

The SAMPLINGTEST, as its name implies, uses a sampling strategy to estimate the number of coordinates with non-negligible variation in a given function  $f$ . The sampling test generates a random subset  $T \subseteq [n]$  of coordinates in each round, and uses the INDEPENDENCETEST to determine if  $f$  is independent of the coordinates in  $T$ . The test accepts when the fraction of rounds that pass the independence test is not much smaller than the expected fraction of rounds that pass the test when  $f$  is a  $k$ -junta. The details of the algorithm are presented in Fig. 2.

**Proposition 3.3 (Completeness).** *Fix  $\eta > 0$ ,  $l \in [k]$ . Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a  $k$ -junta. Then the SAMPLINGTEST accepts  $f$  with probability at least  $1 - \delta$ .*

*Proof.* When  $f$  is a  $k$ -junta, the probability that the set  $T$  in a given round contains only coordinates  $i$  with variation  $\text{Vr}_f(i) = 0$  is at least  $(1 - 1/k)^k$ . When this occurs, the set  $T$  also has variation  $\text{Vr}_f(T) = 0$ . Let  $t$  be the number of rounds for which the set  $T$  satisfies  $\text{Vr}_f(T) = 0$ . By Hoeffding's bound,

$$\Pr \left[ \frac{t}{r} < (1 - 1/k)^k - \frac{l}{16k} \right] \leq e^{-2r \cdot (l/16k)^2} \leq \delta/2$$

when  $r \geq 128k^2 \ln(2/\delta)/l^2$ . Every set  $T$  with variation  $\text{Vr}_f(T) = 0$  always passes the INDEPENDENCETEST, so  $c \geq t$  and the completeness claim follows.  $\square$

**Proposition 3.4 (Soundness 1).** *Fix  $\eta > 0$ ,  $l \in [k]$ . Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a function for which there is a set  $S \subseteq [n]$  of size  $|S| = k + l$  such that every coordinate  $i \in S$  has variation  $\text{Vr}_f(i) \geq \eta$ . Then the SAMPLINGTEST rejects  $f$  with probability at least  $1 - \delta$ .*

*Proof.* In a given round, the probability that the random set  $T$  does not contain any of the  $k + l$  coordinates with large variation is  $(1 - 1/k)^{k+l}$ . When  $l \leq k$ ,  $(1 - 1/k)^l \leq 1 - l/2k$ , and when  $k \geq 2$ ,  $(1 - 1/k)^k \geq 1/4$ . So the probability that

$T$  contains none of the  $k + l$  coordinates with large variation is  $(1 - 1/k)^{k+l} \leq (1 - 1/k)^k(1 - l/2k) \leq (1 - 1/k)^k - l/8k$ .

Let  $t$  represent the number of rounds whose sets  $T$  contain no coordinate with variation at least  $\eta$ . By Hoeffding's bound,

$$\Pr \left[ \frac{t}{r} > ((1 - 1/k)^k - l/8k) + l/16k \right] \leq e^{-2r(l/16k)^2} \leq \delta/2$$

when  $r \geq 128k^2 \ln(2/\delta)/l^2$ . By Fact 2.2 (i), every set  $T$  that contains one of the coordinates  $i$  with variation  $\text{Vr}_f(i) \geq \eta$  also has variation  $\text{Vr}_f(T) \geq \eta$ . By our choice of  $m$ , the probability that the INDEPENDENCETEST accepts a set with variation  $\eta$  is at most  $\delta/2r$ . By the union bound, the INDEPENDENCETEST correctly rejects all the sets with variation at least  $\eta$  except with probability at most  $\delta/2$ .

The sampling test can accept  $f$  only if more than a  $(1 - 1/k)^k - l/16k$  fraction of the random sets contain no coordinate with variation  $\eta$ , or if at least one of those random sets contains such a coordinate but still passes the INDEPENDENCETEST. So the probability that the sampling test erroneously accepts  $f$  is at most  $\delta/2 + \delta/2 = \delta$ .  $\square$

**Proposition 3.5 (Soundness 2).** *Let  $\eta = \frac{\epsilon}{64H_k}$ ,<sup>4</sup> and let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a function for which there exists a set  $S \subseteq [n]$  of coordinates satisfying the following two properties:*

- (i) *Each coordinate  $i \in S$  has variation  $\text{Vr}_f(i) < \eta$ , and*
- (ii) *The total variation of the set  $S$  is  $\text{Vr}_f(S) \geq \epsilon/2$ .*

*Then when  $l = k$ , the SAMPLINGTEST rejects  $f$  with probability at least  $1 - \delta$ .*

The proof of Proposition 3.5 follows very closely the proof of Fischer et al. [10, Lem. 4.3]. In particular, the proof uses the following Chernoff-like bound.

**Lemma 3.6 (Fischer et al. [10, Prop. 3.5]).** *Let  $X = \sum_{i=1}^l X_i$  be a sum of non-negative independent random variables  $X_i$ . If every  $X_i$  is bounded above by  $t$ , then for every  $\lambda > 0$*

$$\Pr[X < \lambda \mathbf{E}[X]] < \exp\left(\frac{\mathbf{E}[X]}{et}(\lambda e - 1)\right).$$

The proof of Proposition 3.5 also makes extensive use of Fischer et al.'s concept of *unique variation* [10].

**Definition 3.7 (Fischer et al. [10]).** *The unique variation of the coordinate  $i \in [n]$  with respect to the set  $S \subseteq [n]$  in the function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is*

$$\text{Ur}_{f,S}(i) = \text{Vr}_f([i] \cap S) - \text{Vr}_f([i-1] \cap S).$$

*Furthermore, the unique variation of the set  $I \subseteq [n]$  of coordinates with respect to  $S$  in  $f$  is  $\text{Ur}_{f,S}(I) = \sum_{i \in I} \text{Ur}_{f,S}(i)$ .*

<sup>4</sup> Recall that  $H_k = \sum_{j=1}^k \frac{1}{j}$  is the  $k$ th harmonic number.

**Fact 3.8 (Fischer et al. [10])** For any function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and sets of coordinates  $S, T \subseteq [n]$ , the following two properties hold:

- (i)  $\text{Ur}_{f,S}(T) \leq \text{Vr}_f(T)$ , and
- (ii)  $\text{Ur}_{f,S}([n]) = \text{Vr}_f(S)$ .

We are now ready to complete the proof of Proposition 3.5.

*Proof (of Proposition 3.5).* There are two ways in which the SAMPLINGTEST can accept  $f$ . The test may accept  $f$  if at least a  $(1 - 1/k)^k - 1/16$  fraction of the random sets  $T$  have variation  $\text{Vr}_f(T) < \eta$ . Alternatively, the test may also accept if some of the sets  $T$  with variation  $\text{Vr}_f(T) \geq \eta$  pass the INDEPENDENCETEST. By our choice of  $m$  and the union bound, this latter event happens with probability at most  $\delta/2$ . So the proof of Proposition 3.5 is complete if we can show that the probability of the former event happening is also at most  $\delta/2$ .

Let  $t$  represent the number of rounds where the random set  $T$  has variation  $\text{Vr}_f(T) \geq \eta$ . We want to show that  $\Pr[t/r \geq (1 - 1/k)^k - 1/16] \leq \delta/2$ . In fact, since  $(1 - 1/k)^k \geq 1/4$  for all  $k \geq 2$ , it suffices to show that  $\Pr[t/r \geq 3/16] \leq \delta/2$ .

In a given round, the expected unique variation of the random set  $T$  with respect to  $S$  in  $f$  is

$$\mathbf{E}[\text{Ur}_{f,S}(T)] = \sum_{i \in [n]} \frac{1}{k} \text{Ur}_{f,S}(i) = \frac{\text{Ur}_{f,S}([n])}{k} = \frac{\text{Vr}_f(S)}{k} \geq \frac{\epsilon}{2k},$$

where the third equality uses Fact 3.8 (ii). By Property (i) of the Proposition,  $\text{Ur}_{f,S}(T)$  is the sum of non-negative variables that are bounded above by  $\eta$ . So we can apply Lemma 3.6 with  $\lambda = 1/32H_k$  to obtain

$$\Pr[\text{Ur}_f(T) < \eta] < e^{\frac{\epsilon}{2\epsilon k \eta} \left( \frac{\epsilon}{32H_k} - 1 \right)}.$$

By Fact 3.8 (i) and the fact that  $e^{\frac{\epsilon}{2\epsilon k \eta} \left( \frac{\epsilon}{32H_k} - 1 \right)} < 1/8$  for all  $k \geq 1$ , we have that

$$\mathbf{E}[t/r] = \Pr[\text{Vr}_f(T) < \eta] < 1/8.$$

The final result follows from an application of Hoeffding's inequality and the choice of  $r$ .  $\square$

The SAMPLINGTEST algorithm makes  $2m$  queries to  $f$  in each round, so the total query complexity of the algorithm is  $2rm = O(k^2 \log(k/l\delta)/l^2\eta)$ .

### 3.3 The Junta Test

In the previous two subsections, we defined two tests: the BLOCKTEST that distinguishes  $k$ -juntas from functions with  $k+1$  coordinates with large variation, and the SAMPLINGTEST that distinguishes  $k$ -juntas from functions that have some variation distributed over a large number of coordinates. The following structural Lemma on functions that are  $\epsilon$ -far from being  $k$ -juntas shows that these two tests are sufficient for testing juntas.



JUNATTEST( $f, k, \epsilon$ )

Additional parameters:  $\delta = \frac{1}{3(\lceil \log k^{1/2} \rceil + 2)}$ ,  $\tau = \frac{\epsilon}{64H_k}$

1. Run BLOCKTEST( $f, k, \tau/\lceil k^{1/2} \rceil, \delta$ ).
2. For  $l = \lceil k^{1/2} \rceil, \lceil 2k^{1/2} \rceil, \lceil 4k^{1/2} \rceil, \lceil 8k^{1/2} \rceil, \dots, k$ ,
  - 2.1. Run SAMPLINGTEST( $f, k, l, \tau/2l, \delta$ ).
3. Run SAMPLINGTEST( $f, k, k, \tau/k, \delta$ ).
4. **Accept**  $f$  if all of the above tests accept; otherwise **reject**  $f$ .

**Fig. 3.** The algorithm for the junta test.

**Lemma 3.9.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be  $\epsilon$ -far from being a  $k$ -junta. Then for any  $t > 0$ ,  $f$  satisfies at least one of the following two properties:*

- (i) *There exists an integer  $l \in [k]$  such that there are at least  $k+l$  coordinates  $i$  with variation  $\text{Vr}_f(i) \geq \frac{\epsilon}{tH_k l}$  in  $f$ .*
- (ii) *The set  $S$  of coordinates  $i \in [n]$  with variation  $\text{Vr}_f(i) < \frac{\epsilon}{tH_k k}$  has total variation  $\text{Vr}_f(S) \geq (1 - 1/t)\epsilon$ .*

*Proof.* Let  $f$  be a function that does not satisfy the Property (i) of the Lemma. Define  $J \subseteq [n]$  to be the set of the  $k$  coordinates in  $f$  with highest variation, and let  $T$  be the set of coordinates  $i \in [n] \setminus J$  with variation  $\text{Vr}_f(i) \geq \frac{\epsilon}{tH_k k}$ . Since  $f$  does not satisfy Property (i) of the Lemma, Fact 2.2 (ii) ensures that the variation of  $T$  is bounded by  $\text{Vr}_f(T) \leq \frac{\epsilon}{tH_k} + \frac{\epsilon}{2tH_k} + \dots + \frac{\epsilon}{ktH_k} = \frac{\epsilon}{t}$ . Since  $S \cup T \supseteq [n] \setminus J$  and any function  $\epsilon$ -far from being a  $k$ -junta must satisfy  $\text{Vr}_f([n] \setminus J) \geq \epsilon$ , a second application of Fact 2.2 (ii) shows that  $f$  must satisfy Property (ii) of the Lemma.  $\square$

Lemma 3.9 naturally suggests an algorithm for testing  $k$ -juntas: use the BLOCKTEST (with parameter  $\eta = \epsilon/64H_k k$ ) to reject functions that satisfy Property (i) of the Lemma, and use the SAMPLINGTEST (with parameters  $l = k$  and  $\eta$  as above) to reject the functions that satisfy Property (ii) of the Lemma. This algorithm is equivalent to the non-adaptive algorithm of Fischer et al. [10], and requires  $\tilde{O}(k^2)/\epsilon$  queries.

We can improve the query complexity of the algorithm by splitting up the task of identifying functions that satisfy Property (i) of Lemma 3.9 into multiple tasks for more specific ranges of  $l$ . The result of this approach is the JUNATTEST algorithm presented in Fig. 3. With this algorithm, we are now ready to prove Theorem 1.1.

**Theorem 1.1.** *The property of being a  $k$ -junta can be  $\epsilon$ -tested by a non-adaptive algorithm with  $\tilde{O}(k^{3/2})/\epsilon$  queries.*

*Proof.* Let us begin by showing that the JUNATTEST is a valid algorithm for  $\epsilon$ -testing  $k$ -juntas. By Propositions 3.1 and 3.3,  $k$ -juntas pass the BLOCKTEST and each of the SAMPLINGTEST instances with probability  $\delta$ . So by our choice of  $\delta$

and the union bound,  $k$ -juntas are accepted by the `JUNTA TEST` with probability at least  $2/3$ .

Let  $f$  be any function that is  $\epsilon$ -far from being a  $k$ -junta. If  $f$  satisfies Property (i) of Lemma 3.9 with parameter  $t = 64$ , consider the minimum integer  $l' \in [k]$  for which there is a set  $S \subseteq [n]$  of size  $k + l'$  such that every coordinate  $i \in S$  has variation  $\text{Vr}_f(i) \geq \frac{\epsilon}{64H_k l'}$ . If  $l' < \lceil k^{1/2} \rceil$ , then by Proposition 3.2, the `BLOCK TEST` rejects  $f$  with probability  $1 - \delta > 2/3$ . If  $l' \geq k^{1/2}$ , then by Proposition 3.4, the `SAMPLING TEST` with the parameter  $l$  that satisfies  $l \leq l' \leq 2l$  rejects the function with probability  $1 - \delta > 2/3$ .

If  $f$  satisfies Property (ii) of Lemma 3.9, by Proposition 3.5, the last `SAMPLING TEST` rejects the function with probability  $1 - \delta > 2/3$ . Since Lemma 3.9 guarantees that any function  $\epsilon$ -far from being a  $k$ -junta must satisfy at least one of the two properties of the Lemma, this completes the proof of soundness of the `JUNTA TEST`.

To complete the proof of Theorem 1.1, it suffices to show that the `JUNTA TEST` is a non-adaptive algorithm and that it makes only  $\tilde{O}(k^{3/2})/\epsilon$  queries to the function. The non-adaptivity of the `JUNTA TEST` is apparent from the fact that all queries to the input function come from independent instances of the `INDEPENDENCE TEST`. The query complexity of the `JUNTA TEST` also follows from the observation that each instance of the `BLOCK TEST` or the `SAMPLING TEST` in the algorithm requires  $\tilde{O}(k^{3/2})/\epsilon$  queries. Since there are a total of  $O(\log k)$  calls to those tests, the total query complexity of the `JUNTA TEST` is also  $\tilde{O}(k^{3/2})/\epsilon$ .  $\square$

## 4 The Lower Bound

In this section, we show that every non-adaptive algorithm for  $\epsilon$ -testing  $k$ -juntas must make at least  $\min(\tilde{\Omega}(k/\epsilon), 2^k/k)$  queries to the function.

To prove Theorem 1.2, we introduce two distributions,  $\mathcal{D}_{yes}$  and  $\mathcal{D}_{no}$ , over functions that are  $k$ -juntas and functions that are  $\epsilon$ -far from  $k$ -juntas with high probability, respectively. We then show that no deterministic non-adaptive algorithm can reliably distinguish between functions drawn from  $\mathcal{D}_{yes}$  and functions drawn from  $\mathcal{D}_{no}$ . The lower bound on all non-adaptive algorithms for  $\epsilon$ -testing  $k$ -juntas then follows from an application of Yao's Minimax Principle [20].

A central concept that we use extensively in the proof of Theorem 1.2 is Chockler and Gutfreund's definition of twins [8].

**Definition 4.1.** *Two vectors  $x, y \in \{0, 1\}^n$  are called  $i$ -twins if they differ exactly in the  $i$ th coordinate (i.e., if  $x_i \neq y_i$  and  $x_j = y_j$  for all  $j \in [n] \setminus \{i\}$ ). The vectors  $x, y$  are called twins if they are  $i$ -twins for some  $i \in [n]$ .*

We now define the distributions  $\mathcal{D}_{yes}$  and  $\mathcal{D}_{no}$ . To generate a function from the distribution  $\mathcal{D}_{no}$ , we first define a function  $g : \{0, 1\}^{k+1} \rightarrow \{0, 1\}$  by setting the value  $g(x)$  for each input  $x \in \{0, 1\}^{k+1}$  independently at random, with  $\Pr[g(x) = 1] = 6\epsilon$ . We then extend the function over the full domain by defining  $f(x) = g(x_{[k+1]})$  for every  $x \in \{0, 1\}^n$ . The distribution  $\mathcal{D}_{yes}$  is defined to be

the uniform mixture distribution over the distributions  $\mathcal{D}_{yes}^{(1)}, \mathcal{D}_{yes}^{(2)}, \dots, \mathcal{D}_{yes}^{(k+1)}$ , where the distribution  $\mathcal{D}_{yes}^{(i)}$  is defined similarly to the  $\mathcal{D}_{no}$  distribution, but over the set  $[k+1] \setminus \{i\}$  instead of  $[k+1]$ .

By construction, the functions drawn from  $\mathcal{D}_{yes}$  are all  $k$ -juntas. The following Lemma shows that a function drawn from  $\mathcal{D}_{no}$  is  $\epsilon$ -far from being a  $k$ -junta with high probability.

**Lemma 4.2.** *When  $k/2^k < \epsilon \leq 1/12$  and  $k \geq 3$ , a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  drawn from  $\mathcal{D}_{no}$  is  $\epsilon$ -far from being a  $k$ -junta with probability at least  $11/12$ .*

*Proof.* A function  $f$  drawn from  $\mathcal{D}_{no}$  is  $\epsilon$ -far from being a  $k$ -junta iff the function  $g : \{0, 1\}^{k+1} \rightarrow \{0, 1\}$  that was extended to form  $f$  is  $\epsilon$ -far from being a  $k$ -junta. In turn,  $g$  is  $\epsilon$ -far from being a  $k$ -junta iff for every coordinate  $i \in [k+1]$ , we must change the value of  $g(x)$  on at least  $\epsilon 2^{k+1}$  different inputs  $x \in \{0, 1\}^{k+1}$  to make the function  $g$  independent of the  $i$ th variable – which is equivalent to requiring that at least  $\epsilon 2^{k+1}$  pairs of  $i$ -twins have distinct values in  $g$ .

Consider a fixed  $i \in [k+1]$ . Since each value  $g(x)$  is generated independently and takes value  $g(x) = 1$  with probability  $6\epsilon$ , each pair of  $i$ -twins has distinct values with probability  $2 \cdot 6\epsilon(1 - 6\epsilon)$ . Let  $t_i$  represent the number of  $i$ -twins with distinct values in  $g$ . Then when  $\epsilon \leq 1/12$ ,  $\mathbf{E}[t_i] = 12\epsilon(1 - 6\epsilon)2^k \geq 6\epsilon 2^k$ , and we can apply Chernoff's bound to obtain

$$\Pr [t_i \leq \epsilon 2^{k+1}] \leq e^{-6\epsilon 2^k (1-1/3)^2/2} = e^{-\epsilon 2^{k+2}/3}.$$

The Lemma then follows from the union bound and the conditions that  $\epsilon > k/2^k$  and  $k \geq 3$ .  $\square$

Consider any sequence of  $q$  queries that a deterministic non-adaptive algorithm may make to a function  $f$ . We want to show that when  $q$  is small, the responses observed by the algorithm when  $f$  is drawn from  $\mathcal{D}_{yes}$  are very similar to the responses observed when  $f$  is drawn from  $\mathcal{D}_{no}$ . The following Lemma provides a first step toward that goal.

**Lemma 4.3.** *Let  $Q$  be a set of  $q$  queries containing  $t_i$   $i$ -twins. Let  $\mathcal{R}_{yes}^{(i)}$  and  $\mathcal{R}_{no}$  be the distributions of the responses to the queries in  $Q$  when the input function is drawn from  $\mathcal{D}_{yes}^{(i)}$  or  $\mathcal{D}_{no}$ , respectively. Then the statistical distance between  $\mathcal{R}_{yes}^{(i)}$  and  $\mathcal{R}_{no}$  is bounded above by*

$$\sum_{y \in \{0, 1\}^q} |\mathcal{R}_{yes}^{(i)}(y) - \mathcal{R}_{no}(y)| \leq 24t_i\epsilon.$$

*Proof.* We apply a hybridization argument. Let the pairs of  $i$ -twins in  $Q$  be represented by  $(\alpha_1, \beta_1), \dots, (\alpha_{t_i}, \beta_{t_i})$ . For  $j \in \{0, 1, \dots, t_i\}$ , define the response distribution  $\mathcal{H}_j$  to be the distribution where each response is independent and  $6\epsilon$ -biased, except for the responses  $\beta_1, \dots, \beta_j$ , which are constrained to satisfy  $\alpha_1 = \beta_1, \dots, \alpha_j = \beta_j$ . Note that  $\mathcal{H}_0 = \mathcal{R}_{no}$  and  $\mathcal{H}_{t_i} = \mathcal{R}_{yes}^{(i)}$ , so

$$\sum_y |\mathcal{R}_{yes}^{(i)}(y) - \mathcal{R}_{no}(y)| = \sum_y |\mathcal{H}_{t_i}(y) - \mathcal{H}_0(y)| \leq \sum_{j=1}^{t_i} \sum_y |\mathcal{H}_j(y) - \mathcal{H}_{j-1}(y)|.$$

The distributions  $\mathcal{H}_j$  and  $\mathcal{H}_{j-1}$  are nearly identical. The only difference between the two distributions is that  $\beta_j$  is constrained to take the value  $\alpha_j$  in  $\mathcal{H}_j$ , while it is an independent  $6\epsilon$ -biased random variable in  $\mathcal{H}_{j-1}$ . So the statistical distance between  $\mathcal{H}_j$  and  $\mathcal{H}_{j-1}$  is twice the probability that  $\beta_j \neq \alpha_j$  in  $\mathcal{H}_{j-1}$ . Thus,  $\sum_y |\mathcal{H}_j(y) - \mathcal{H}_{j-1}(y)| \leq 24\epsilon(1 - 6\epsilon) < 24\epsilon$  and the Lemma follows.  $\square$

With Lemma 4.3, we can now bound the statistical distance between the responses observed when the input function is drawn from  $\mathcal{D}_{yes}$  or  $\mathcal{D}_{no}$ .

**Lemma 4.4.** *Let  $Q$  be a sequence of  $q$  queries containing  $t$  pairs of twins. Let  $\mathcal{R}_{yes}$  and  $\mathcal{R}_{no}$  be the distributions of the responses to the queries in  $Q$  when the input function is drawn from  $\mathcal{D}_{yes}$  or  $\mathcal{D}_{no}$ , respectively. Then*

$$\sum_{y \in \{0,1\}^q} |\mathcal{R}_{yes}(y) - \mathcal{R}_{no}(y)| \leq \frac{24t\epsilon}{k+1}.$$

*Proof.* Since  $\mathcal{R}_{yes}$  is a mixture distribution over  $\mathcal{R}_{yes}^{(1)}, \dots, \mathcal{R}_{yes}^{(k+1)}$ , then

$$\sum_y |\mathcal{R}_{yes}(y) - \mathcal{R}_{no}(y)| = \sum_y \left| \sum_{i=1}^{k+1} \frac{\mathcal{R}_{yes}^{(i)}(y)}{k+1} - \mathcal{R}_{no}(y) \right| \leq \frac{1}{k+1} \sum_{i=1}^{k+1} \sum_y |\mathcal{R}_{yes}^{(i)}(y) - \mathcal{R}_{no}(y)|.$$

By Lemma 4.3, the above equation is upper bounded by  $\frac{1}{k+1} \sum_{i=1}^{k+1} 24t_i\epsilon$ , where  $t_i$  represents the number of  $i$ -twins in  $Q$ . Lemma 4.4 then follows from the fact that  $t = \sum_{i=1}^{k+1} t_i$ .  $\square$

The previous Lemma bounds the statistical distance between the responses observed from a function drawn from  $\mathcal{D}_{yes}$  or  $\mathcal{D}_{no}$  when we have a bound on the number of twins in the queries. The following Lemma shows that the number of pairs of twins in a sequence of  $q$  queries can not be larger than  $q \log q$ .

**Lemma 4.5.** *Let  $\{x_1, \dots, x_q\} \subseteq \{0,1\}^n$  be a set of  $q$  distinct queries to a function  $f : \{0,1\}^n \rightarrow \{0,1\}$ . Then there are at most  $q \log q$  pairs  $(x_i, x_j)$  such that  $x_i$  and  $x_j$  are twins.*

*Proof.* A natural combinatorial representation for a query  $x \in \{0,1\}^n$  is as a vertex on the  $n$ -dimensional boolean hypercube. In this representation, a pair of twins corresponds to a pair of vertices connected by an edge on the hypercube. So the number of pairs of twins in a set of queries is equal to the number of edges contained in the corresponding subset of vertices on the hypercube. The Lemma then follows from the Edge-Isoperimetric Inequality of Harper [13], Bernstein [3], and Hart [14] (see also [7, §16]), which states that any subset  $S$  of  $q$  vertices in the boolean hypercube contains at most  $q \log q$  edges.<sup>5</sup>  $\square$

We can now combine the above Lemmas to prove Theorem 1.2.

<sup>5</sup> The result of Harper, Bernstein, and Hart is slightly tighter, giving a bound of  $\sum_{i=1}^q h(i)$ , where  $h(i)$  is the number of ones in the binary representation of  $i$ .

**Theorem 1.2.** *Any non-adaptive algorithm for  $\epsilon$ -testing  $k$ -juntas must make at least  $\min\left(\Omega\left(\frac{k/\epsilon}{\log k/\epsilon}\right), \Omega\left(\frac{2^k}{k}\right)\right)$  queries.*

*Proof.* Let us first consider the case where  $\epsilon \geq k/2^k$ . Let  $\mathcal{A}$  be any non-adaptive deterministic algorithm for testing  $k$ -juntas with  $q = \frac{k/600\epsilon}{\log k/600\epsilon}$  queries. By Lemma 4.5, there can be at most  $q \log q = \frac{k}{600\epsilon}$  pairs of twins in the  $q$  queries. By Lemma 4.4, this means that the statistical distance between the response distributions  $\mathcal{R}_{yes}$  and  $\mathcal{R}_{no}$  is at most  $\frac{k}{600\epsilon} \cdot \frac{24\epsilon}{k+1} < \frac{1}{25}$ . So the algorithm  $\mathcal{A}$  can not predict which distribution generated a given input with accuracy greater than  $\frac{1}{2} + \frac{1}{2} \cdot \frac{1}{25} = \frac{26}{50}$ . By Lemma 4.2, a function drawn from  $\mathcal{D}_{no}$  fails to be  $\epsilon$ -far from being a  $k$ -junta with probability at most  $\frac{1}{12}$ . So the success rate of  $\mathcal{A}$  is at most  $\frac{26}{50} + \frac{1}{12} < \frac{2}{3}$ . Therefore, by Yao's Minimax Principle, any algorithm for  $\epsilon$ -testing  $k$ -juntas requires  $\Omega\left(\frac{k/\epsilon}{\log k/\epsilon}\right)$  queries.

When  $\epsilon < k/2^k$ , we can repeat the above argument with  $\epsilon' = k/2^k$  instead of  $\epsilon$ . This yields a lower bound of  $\Omega\left(\frac{k/\epsilon'}{\log k/\epsilon'}\right) = \Omega\left(\frac{2^k}{k}\right)$  queries.  $\square$

## 5 Conclusion

Our results have improved the upper bound for the query complexity for testing juntas and the lower bound for testing juntas with non-adaptive algorithms. The results stated in this article are all presented in the context of testing functions with boolean domains, but we note that the results also generalize to the context of testing of functions  $f : X^n \rightarrow \{0, 1\}$  for any finite domain  $X$ .

The results also suggest some interesting problems for future work.

**Open Problem 5.1** *What is the query complexity of the junta testing problem? In particular, can we  $\epsilon$ -test  $k$ -juntas non-adaptively with  $\tilde{O}(k/\epsilon)$  queries?*

Open Problem 5.1 has some relevance to the study of quantum algorithms in property testing: while Theorem 1.1 improves on all known upper bounds for the query complexity of classical algorithms for testing juntas, it still does not match the query complexity of  $O(k/\epsilon)$  obtained by Atıcı and Servedio [1] for a non-adaptive algorithm with access to quantum examples.

**Open Problem 5.2** *Is there a gap between the query complexity of adaptive and non-adaptive algorithms for testing juntas?*

Gonen and Ron [11] showed that such a gap exists for some property testing problems in the dense graph model. A positive answer to Open Problem 5.2 would provide an interesting example of a similar gap in the context of testing function properties.

**Open Problem 5.3** *Can improved query bounds for testing juntas yield better bounds for testing other properties of boolean functions?*

The work of Diakonikolas et al. [9] strongly suggests a positive answer to Open Problem 5.3, since the junta test plays a central role in their generic algorithm for testing many properties of boolean functions.

**Acknowledgments.** The author wishes to thank Ryan O’Donnell for many valuable discussions and suggestions during the course of this research. The author also thanks Anupam Gupta, Yi Wu, and the anonymous referees for many helpful suggestions on earlier drafts of this article.

## References

1. Alp Atıcı and Rocco A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007.
2. Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs and non-approximability – towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.
3. Arthur J. Bernstein. Maximally connected arrays on the  $n$ -cube. *SIAM J. Appl. Math.*, 15(6):1485–1489, 1967.
4. Avrim Blum. Relevant examples and relevant features: thoughts from computational learning theory. In *AAAI Fall Symposium on ‘Relevance’*, 1994.
5. Avrim Blum. Learning a function of  $r$  relevant variables. In *Proc. 16th Conference on Computational Learning Theory*, pages 731–733, 2003.
6. Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(2):245–271, 1997.
7. Béla Bollobás. *Combinatorics*. Cambridge, 1986.
8. Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.
9. Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *Proc. 48th Symposium on Foundations of Computer Science*, pages 549–558, 2007.
10. Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004.
11. Mira Gonen and Dana Ron. On the benefits of adaptivity in property testing of dense graphs. In *Proc. 11th Workshop RANDOM*, pages 525–539, 2007.
12. David Gujarro, Jun Tarui, and Tatsuie Tsukiji. Finding relevant variables in PAC model with membership queries. In *Proc. 10th Conference on Algorithmic Learning Theory*, pages 313–322, 1999.
13. Larry H. Harper. Optimal assignments of numbers to vertices. *SIAM J. Appl. Math.*, 12(1):131–135, 1964.
14. Sergiu Hart. A note on the edges of the  $n$ -cube. *Disc. Math.*, 14:157–163, 1976.
15. Jeff Kahn, Gil Kalai, and Nati Linial. The influence of variables on boolean functions. In *Proc. 29th Sym. on Foundations of Computer Science*, pages 68–80, 1988.
16. 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 *Proc. 20th Conference on Computational Complexity*, pages 112–119, 2005.
17. Elchanan Mossel, Ryan O’Donnell, and Rocco A. Servedio. Learning functions of  $k$  relevant variables. *J. Comput. Syst. Sci.*, 69(3):421–434, 2004.
18. Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM J. Discret. Math.*, 16(1):20–46, 2003.
19. Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
20. Andrew C. Yao. Probabilistic computations: towards a unified measure of complexity. In *Proc. 18th Sym. on Foundations of Comput. Sci.*, pages 222–227, 1977.