A Lower Bound for Testing Juntas

Hana Chockler and Dan Gutfreund
School of Engineering and Computer Science
Hebrew University
Jerusalem 91904, Israel.
Email: {hanac,danig}@cs.huji.ac.il

October 31, 2002

Abstract

We show an \( \Omega(m) \) lower bound on the number of queries required to test whether a Boolean function depends on at most \( m \) out of its \( n \) variables. This improves a previously known lower bound for testing this property. Our proof is simple and uses only elementary techniques.

Keywords

Property testing, algorithms, randomized algorithms, computational complexity, Boolean functions.

1 Introduction

Property testing was first introduced by Rubinfeld and Sudan [RS96], who checked whether a given function computes a low-degree polynomial or is far from computing it. Their work has led to the study of combinatorial property testing, defined by Goldreich et al. [GGR98]. Generally speaking, given a property \( P \), an input \( x \), and \( 0 < \epsilon < 1 \), we say that \( x \) is \( \epsilon \)-far from satisfying \( P \) if we need to change at least \( \epsilon \)-fraction of the bits in the description of \( x \) in order for it to satisfy \( P \). For a given property \( P \) and \( 0 < \epsilon < 1 \), a randomized algorithm \( A \) is an \( (\epsilon, q) \)-test for \( P \) if \( A \) queries at most \( q \) bits of the input and distinguishes with probability at least \( 2/3 \) between inputs that have the property \( P \) and inputs that are \( \epsilon \)-far from having the property \( P \). The algorithm is not required to answer correctly on inputs that neither have the property \( P \), nor are \( \epsilon \)-far from having the property \( P \). The efficiency of the algorithm is usually measured by its query complexity (i.e. the number of bits that \( A \) reads from the input).

Most of the previous research on property testing focus on testing graph properties. There exist efficient testing algorithms for several graph properties [GR97, GGR98, GR99, BR00, AK], as well as lower bounds on the query complexity [GR97, BR00]. Other areas that were studied include the testability of classes of language (e.g. regular and context-free languages) [AKNS99, PRR01], algebraic properties of functions [BLR93], and Boolean functions [GGL+00, PRS01, FKR+02]. In all these areas both algorithms and lower bounds where obtained.

In this paper we focus on testing properties of Boolean functions. Specifically, we study the property of a Boolean function depending on at most \( m \) (out of \( n \)) of its variables (we call this property \( m \)-dependence), where \( m \) can be a function of the total number of variables \( n \).
The dependence of Boolean functions on a constant $k$ of the variables (aka $k$-juntas) was studied by Fischer et al. [FKR+02]. They showed a testing algorithm for this property that has 1-sided error and query complexity $O(k^3)$ and an algorithm that has two-sided error and query complexity $O(k^4)$. They also proved a lower bound of $\Omega(\sqrt{k})$ for algorithms that are restricted to ask their queries non-adaptively (from this they deduced a $\Omega(\log k)$ lower bound for adaptive algorithms). The lower bound of [FKR+02] applies to $k = o(\sqrt{n})$.

We show an $\Omega(\tau_m)$ lower bound on the query complexity of testing whether a Boolean function is $m$-dependent ($m$ can be any function of the total number of variables), thus improving the $\Omega(\sqrt{k})$ lower bound of [FKR+02] (for a constant $k$). Our lower bound is not restricted to algorithms that make non-adaptive queries, and works for all $m < n$. Furthermore, its proof uses only elementary analysis.

## 2 Preliminaries

### 2.1 Definitions and Notations

An input of length $n$ is a vector of $n$ bits. The distance $\text{dist}(x, y)$ between two vectors $x$ and $y$ of the same length is the Hamming distance between them, that is, the number of bits that have to be changed in $x$ to obtain $y$. For $0 < \varepsilon \leq 1$, a vector $x$ of length $n$ is $\varepsilon$-far from a vector $y$ if $\text{dist}(x, y) \geq \varepsilon n$. For a property $P$, a vector $x$ is $\varepsilon$-far from having the property $P$ if $x$ is $\varepsilon$-far from all vectors that have the property $P$. The following definition of property testing is due to [GGR98].

**Definition 2.1 (Property Testing) [GGR98]** Let $P$ be a fixed property. Let $A$ be a probabilistic algorithm which receives as input a size parameter $n$ and a distance parameter $0 < \varepsilon \leq 1$. The algorithm $A$ is also given access to an oracle for the input $x$ of size $n$ (that is, $A$ can query the $i$-th bit of $x$ in time $O(1)$, for each $0 \leq i \leq n - 1$). We say that $A$ is an $(\varepsilon, q)$-test for the property $P$ if for every input $x$ of size $n$, $A$ makes at most $q$ queries, and the following holds.

- If $x$ has the property $P$, then $A$ accepts with probability at least $2/3$.
- If $x$ is $\varepsilon$-far from having the property $P$ then $A$ rejects with probability at least $2/3$.

In this paper we consider properties of Boolean functions. An input function $f$ on $n$ variables is represented by its $2^n$-long truth table. We are interested in testing the number of variables that influence the value of an input function. The $m$-dependence property is defined as follows.

**Definition 2.2** A function $f$ is $m$-dependent if $f$ depends on at most $m$ of its variables, where $m = m(n) < n$.

For example, for $m = n - 1$ the $m$-dependence property is: “the input function does not depend on all its variables”, and for $m = n/2$, the $m$-dependence property is: “the input function does not depend on a majority of its variables”.

For $s \in \{0, 1\}^n$ and an index $1 \leq i \leq n$, we say that $t \in \{0, 1\}^n$ is the $i$-twin of $s$ if $\text{dist}(s, t) = 1$, and they differ only in the $i$-th bit.

For a distribution $D$, we denote by $x \leftarrow D$ a sample $x$ taken from $D$. 
3 The lower bound

In this section we prove a lower bound of \( \Omega(m) \) on the query complexity of algorithms that test \( m \)-dependence for all \( m = m(n) < n \). We start with the following lemma.

**Lemma 3.1** Let \( D_{n,m} \) (for \( m = m(n) < n \)) be the uniform distribution over all Boolean functions that do not depend on the last \( n - m \) variables. Then for \( 0 < \varepsilon < 1/2 \) w.h.p. a function \( f \), chosen from the distribution \( D_{n,m} \), is \( \varepsilon \)-far from depending on less than \( m \) variables.

**Proof:** A function \( f \) on \( n \) variables that depends on its first \( m \) variables is \( \varepsilon \)-far from depending on less variables iff for each variable \( x_i \) (\( 1 \leq i \leq m \)), we need to change at least \( \varepsilon 2^n \) bits in the truth table of \( f \) to make it independent of \( x_i \). In other words, \( f \) is \( \varepsilon \)-far from not depending on \( m \) of its variables iff in the set of \( 2^n \) possible inputs, for each \( 1 \leq i \leq m \) there are at least \( \varepsilon 2^n \) pairs of \( i \)-twins with different values of \( f \).

Let us first fix an index \( 1 \leq i \leq m \) and compute the probability of \( f \leftarrow D_{n,m} \) being \( \varepsilon \)-far from not depending on \( x_i \). Each pair of \( i \)-twins has different values with probability \( 1/2 \), and thus the expected number of pairs of \( i \)-twins with different values is half the number of pairs, i.e. \( 2^{n-2} \). Let \( t_i \) be the number of \( i \)-twins that have different values of \( f \). By the Chernoff bound [MR95],

\[
\Pr(t_i \leq \varepsilon 2^n) = \Pr[|t_x - 2^{n-2}| \geq \frac{1}{2} - 2\varepsilon 2^{n-1}] \leq \exp(-\theta(2^{2n})).
\]

By the union bound, the probability that a function \( f \) chosen from \( D_{n,m} \) is not \( \varepsilon \)-far from depending on less than \( m \) of its variables is bounded by \( m \times \exp(-\theta(2^{2n})) \).

We can now prove the main theorem.

**Theorem 3.2** Let \( m = m(n) < n \) and \( 0 < \varepsilon < 1/2 \). Let \( A \) be an \( (\varepsilon, l) \)-test for \( m \)-dependence. Then \( A \) makes \( \Omega(m) \) queries.

**Proof:** The technique we use to prove the lower bound was previously used in several proofs of lower bounds for property testing algorithms (see, for example, [GR97, BR00]).

Consider the following distributions over Boolean functions on \( n \) variables:

1. \( D_{n,m}^0 \) is the uniform distribution over all the functions on \( n \) variables that do not depend on the last \( n - (m + 1) \) variables.

2. \( D_{n,m}^1 \) is the distribution obtained in the following way. An index \( i \) is chosen uniformly from \([1..m+1] \), and then a function \( f \) is chosen uniformly among functions that do not depend on the last \( n - (m + 1) \) variables and the variable \( x_i \).

By Lemma 3.1, w.h.p. \( f \leftarrow D_{n,m}^0 \) is \( \varepsilon \)-far from depending on at most \( m \) variables. On the other hand, by definition, every \( f \leftarrow D_{n,m}^1 \) depends on at most \( m \) of its variables.

Let \( A \) be a randomized algorithm for testing \( m \)-dependence using \( l = l(n) \) queries. Namely, \( A \) is a probabilistic mapping from query-answer histories \( [(q_1, a_1), \ldots, (q_i, a_i)] \) to \( q_{i+1} \) for every \( t < l \), and to \{accept, reject\} for \( t = l \). A query \( q_t \) is a Boolean string of \( n \) bits. We assume that the mapping
is defined only on histories which are consistent with some Boolean function (that is, if \( q_i = q_j \) for some \( i \neq j \), then \( a_i = a_j \)). A query-answer history of length \( t - 1 \) can be used to define a knowledge function \( f_{t-1} \) at time \( t - 1 \) (before the \( t \)-th query), where the truth table of \( f_{t-1} \) is known only for inputs \( q_1, \ldots, q_{t-1} \).

In what follows we describe two random processes \( P_0 \) and \( P_1 \), which interact with an arbitrary algorithm \( \mathcal{A} \), so that for \( j \in \{0, 1\} \), \( P_j \) answers the queries of \( \mathcal{A} \) while constructing a random function from \( D_{n,m}^l \). For a fixed \( \mathcal{A} \) that uses \( l \) queries, and for \( j \in \{0, 1\} \), let \( C_i^A \) denote the distribution over query-answer histories of length \( l \), induced by the interaction of \( \mathcal{A} \) and \( P_j \). We show that for any given \( \mathcal{A} \) that uses less than \( (m + 1)/3 \) queries, the statistical difference between \( C_0^A \) and \( C_1^A \) is at most 1/3. The lower bound then follows.

The process \( P_0 \) has two stages. In the first stage the process answers the queries of \( \mathcal{A} \). In the second stage the process completes the knowledge function to a random function from \( D_{n,m}^l \). In the first stage \( P_0 \) answers 0 or 1 with equal probability on each query of \( \mathcal{A} \), unless \( \mathcal{A} \) asked previously a query that gave the same values to the set of variables \( \{x_1, \ldots, x_{m+1}\} \). In this case it gives the same answer. In the second stage, \( P_0 \) uniformly selects a function \( f \) that does not depend on its last \( n - (m + 1) \) variables, and is consistent with the final knowledge function \( f_l \).

The process \( P_1 \) uniformly selects \( i \in \{1, \ldots, m+1\} \). Then it also has two stages. In the first stage, \( P_1 \) answers 0 or 1 with equal probability on each query of \( \mathcal{A} \), unless \( \mathcal{A} \) asked previously a query that gave the same values to the set of variables \( \{x_1, \ldots, x_{m+1}\} \setminus \{x_i\} \). In this case it gives the same answer. In the second stage, \( P_1 \) uniformly selects a function \( f \) that does not depend on its last \( n - (m + 1) \) variables as well as \( x_i \), and is consistent with the final knowledge function \( f_l \).

We can simplify the task of \( \mathcal{A} \) by letting it know that none of its inputs depend on the last \( n - (m + 1) \) variables. Clearly, this can only reduce the number of queries \( \mathcal{A} \) performs. In this case we can assume that in each query, \( \mathcal{A} \) sets the variables \( x_{m+2}, \ldots, x_n \) to zero. We can also assume that \( \mathcal{A} \) never asks the same query twice. We can further simplify the task of \( \mathcal{A} \), by requiring that \( P_1 \) answers “success” in case it is asked about an \( i \)-twin of a query that is currently present in the query-answer history (recall that \( i \) is the index that \( P_1 \) chooses in the first stage).

It is easy to see that for every algorithm \( \mathcal{A} \), the process \( P_0 \) (\( P_1 \)), when interacting with \( \mathcal{A} \), generates the distribution \( D_{n,m}^0 (D_{n,m}^1) \). By construction (and our simplifying assumptions), when \( \mathcal{A} \) interacts with \( P_0 \), for every \( t \leq l \), the bit \( a_t \) is distributed uniformly and independently. When \( \mathcal{A} \) interacts with \( P_1 \), for every \( t \leq l \), if \( q_t \) is not an \( i \)-twin of \( q_j \) for some \( j < l \), then the bit \( a_t \) is distributed uniformly and independently. On the other hand, if for some \( t \leq l \), \( q_t \) is an \( i \)-twin of \( q_j \) for some \( j < l \), then \( P_1 \) answers “success”. So conditioned on the event that the sequence of \( l \) answers that \( \mathcal{A} \) receives from \( P_1 \) does not contain “success”, the distribution over query-answer histories when \( \mathcal{A} \) interacts with \( P_1 \) is identical to the equivalent distribution when \( \mathcal{A} \) interacts with \( P_0 \). Thus, an upper bound on the probability that \( \mathcal{A} \) receives “success” at some stage when it interacts with \( P_1 \), is an upper bound on the statistical difference between \( C_0^A \) and \( C_1^A \). Clearly, \( \mathcal{A} \) cannot gain any information that allows to distinguish between \( P_0 \) and \( P_1 \) from answers to queries that do not contain pairs of \( k \)-twins, for some \( 1 \leq k \leq m + 1 \). If \( \mathcal{A} \) queries a pair of \( k \)-twins, it succeeds to recognize \( P_1 \) only if \( k = i \) (where \( i \) is the index that \( P_1 \) chose in the first stage). In other cases, the answers of \( P_0 \) and \( P_1 \) are indistinguishable. So it all boils down to the question of what is the probability to guess, given \( l \) tries, a number that is chosen uniformly from \( \{1, \ldots, m+1\} \). A simple analysis shows that this probability is at most \( l/(m+1) \). So when \( l < (m+1)/3 \), the statistical difference between \( C_0^A \) and \( C_1^A \) is less than 1/3. \( \square \)
4 Concluding Remarks

In this paper we show an $\Omega(m)$ lower bound for testing $m$-dependence, thus improving the lower bound of [FKR+02]. We prove this lower bound by studying random functions that depend on a fraction of their variables. When proving lower bounds in general and in the field of property testing in particular, the natural inclination is to study special cases that intuitively seem to be the hardest. Indeed, [FKR+02] proved their lower bound by studying parity functions. The proof of Theorem 3.2 illustrates the fact that a random function can be hard enough to obtain strong lower bounds. The advantage of looking at the “random” case is that usually its analysis is much simpler than (specific) hard cases.

Acknowledgment We thank Dana Ron and Alex Samorodnitsky for helpful discussions.

References


