On Approximating the Number of Relevant Variables in a Function

Dana Ron*
School of EE
Tel-Aviv University
Ramat Aviv, Israel
danar@eng.tau.ac.il

Gilad Tsur
School of EE
Tel-Aviv University
Ramat Aviv, Israel
gilad.tsur@gmail.com

July 19, 2011

Abstract

In this work we consider the problem of approximating the number of relevant variables in a function given query access to the function. Since obtaining a multiplicative factor approximation is hard in general, we consider several relaxations of the problem. In particular, we consider a relaxation of the property testing variant of the problem and we consider relaxations in which we have a promise that the function belongs to a certain family of functions (e.g., linear functions). In the former relaxation the task is to distinguish between the case that the number of relevant variables is at most k, and the case in which it is far from any function in which the number of relevant variable is more than $(1+\gamma)k$ for a parameter γ. We give both upper bounds and almost matching lower bounds for the relaxations we study.

*This work was supported by the Israel Science Foundation (grant number 246/08).

1 Introduction

In many scientific endeavors, an important challenge is making sense of huge datasets. In particular, when trying to make sense of functional relationships we would like to know or estimate the number of variables that a function depends upon. This can be useful both as a preliminary process for machine learning and statistical inference and independently, as a measure of the complexity of the relationship in question. We mainly focus on Boolean functions over the Boolean hypercube, which is endowed with the uniform distribution. In the last section we discuss extensions to other finite domains and ranges (as well as other product distributions).

For a function $f : \{0,1\}^n \to \{0,1\}$, we let $r(f)$ denote the number of variables that f depends on, which we shall also refer to as the number of *relevant* variables. A variable x_i is *relevant* to a function f if there exists an assignment to the input variables such that changing the value of just the variable x_i causes the value of f to change. Given query access to f, computing $r(f)$ *exactly* may require a number of queries that is exponential in n (linear in the size of the domain).[1]

Thus, we would like to consider relaxed notions of this computational task. One natural relaxation is to compute $r(f)$ approximately. Namely, to output a value $\hat{r}$ such that $r(f)/B \leq \hat{r} \leq B \cdot r(f)$ for some approximation factor B. Unfortunately, this relaxed task may still require an exponential number of queries (see the example in Footnote 1).

A different type of relaxation that has been studied in the past, is the one defined by property testing [14, 9]. We shall say that f is a *k-junta* if $r(f) \leq k$. A property testing algorithm is given k and a distance parameter $0 < \epsilon < 1$. By performing queries to f, the algorithm should distinguish between the case that f is a k-junta and the case that it differs from every k-junta on at least an ϵ-fraction of the domain (in which case we shall say that it is *ϵ-far* from being a k-junta). This problem was studied in several papers [7, 6, 2, 3]. The best upper bound on the number of queries that the algorithm performs (in terms of the dependence on k) is $O(k \log k)$ [3], where this upper bound almost matches the lower bound of $\Omega(k)$[6].

A natural question, which was raised in [7], is whether it is possible to reduce the complexity below $\tilde{O}(k)$ if we combine the above two relaxations. Namely, we consider the following problem: *Given parameters $k \geq 1$ and $0 < \epsilon, \gamma < 1$ and query access to a function f, distinguish (with high constant probability) between the case that f is a k-junta and the case that f is ϵ-far from any $(1+\gamma)k$-junta.*[2] This problem was recently considered by Blais et al. [4]. They apply a general new technique that they develop for obtaining lower bounds on property testing problems via communication complexity lower bounds. Specifically, they give a lower bound of $\Omega\left(\min\{(\frac{k}{t})^2, k\} - \log k\right)$ on the number of queries necessary for distinguishing between functions that are k-juntas and functions that are ϵ-far from $(k+t)$-juntas (for a constant ϵ). Using our formulation, this implies that we cannot go below a linear dependence on k for $\gamma = O(1/\sqrt{k})$.

Our Results. What if we allow γ to be a constant (i.e., independent of k), say, $\gamma = 1$? Our first main result is that even if we allow γ to be a constant, then the testing problem does not become much easier. Specifically, we prove:

[1] Consider for example the family of functions, where each function in the family takes the value 0 on all points in the domain but one. Such a function depends on all n variables, but a uniformly selected function in the family cannot be distinguished (with constant probability) from the all-0 function, which depends on 0 variables.

[2] We note that problems in the spirit of this problem (which allow a further relaxation to that defined by "standard" property testing) have been studied in the past (e.g., [11, 10, 1]).

Theorem 1.1 *Any algorithm that distinguishes between the case that f is a k-junta and the case that f is ϵ-far from any $(1+\gamma)k$-junta for constant ϵ and γ must perform $\Omega(k/\log(k))$ queries.*

While Theorem 1.1 does not leave much place for improvement of the query complexity as compared to the $O(k \log k)$ upper bound [3] for the standard property testing problem (i.e., when $\gamma = 0$), we show that a small improvement (in terms of the dependence on k) can be obtained:

Theorem 1.2 *There exists an algorithm that, given query access to $f : \{0,1\}^n \to \{0,1\}$ and parameters $k \geq 1$, and $0 < \epsilon, \gamma < 1$, distinguishes with high constant probability between the case that f is a k-junta and the case that f is ϵ-far from any $(1+\gamma)k$-junta. The algorithm performs $O\left(\frac{k \log(1/\gamma)}{\epsilon \gamma^2}\right)$ queries.*

Given that the relaxed property testing problem is not much easier than the standard one in general, we consider another possible relaxation: Computing (approximately) the number of relevant variables of *restricted* classes of functions. For example, suppose that we are given the promise that f is a *linear* function. Since it is possible to exactly learn f (with high constant probability) by performing $O(r(f) \log n)$ queries, it is also possible to exactly compute $r(f)$ in this case using this number of queries. On the other hand, Blais et al. [4] show that in order to distinguish (with constant success probability) between the case that a linear function has k relevant variables and the case that it has more than $k+1$ relevant variables, requires $\Omega(\min\{k, n-k\})$ queries[3] (so that $\Omega(r(f))$ queries are necessary for exactly computing $r(f)$). However, if we allow a constant multiplicative gap, then we get the following result:

Theorem 1.3 *Given query access to a linear function f, it is possible to distinguish with high constant probability between the case that f has at most k relevant variables and the case that f has more than $(1+\gamma)k$ relevant variables by performing $\Theta(\log(1/\gamma)/\gamma^2)$ queries.*

By standard techniques, Theorem 1.3 implies that we can obtain (with high constant probability) a multiplicative approximation of $(1+\gamma)$ for $r(f)$ (when f is a linear function), by performing $\tilde{O}(\log(r(f))/\gamma^2)$ queries to f.

Theorem 1.3, which deals with linear functions, extends to polynomials:

Theorem 1.4 *There exists an algorithm that distinguishes between polynomials of degree at most d with at most k relevant variables and polynomials of degree at most d that have at least $(1+\gamma)k$ relevant variables by performing $O\left(\frac{2^d \log(1/\gamma)}{\gamma^2}\right)$ queries.*

Compared to Theorem 1.2, Theorem 1.4 gives a better result for degree-d polynomials when $d < \log(k)$. A natural question is whether in this case we can do even better in terms of the dependence on d. We show that it is not possible to do much better (even if we also allow the property testing relaxation):

Theorem 1.5 *For fixed values of ϵ (for sufficiently small ϵ), and for $d < \log(k)$, any algorithm that distinguishes between polynomials of degree d with k relevant variables and those that are ϵ-far from all degree-d polynomials with $2k$ relevant variables must perform $\Omega(2^d/d)$ queries.*

[3] A slightly weaker bound of $\Omega(k/\text{polylog}(k))$ was proved independently by Chakraborty et al. [5] based on work by Goldriech [8]).

Finally we show that a lower lower bound similar to the one stated in Theorem 1.1 holds when we have a promise that the function is monotone (except that it holds for $\epsilon = O(1/\log(k))$ rather than constant ϵ).

TECHNIQUES. Our lower bounds build on reductions from the *Distinct Elements* problem: Given query access to a sequence of length n, the goal is approximate the number of *distinct* elements in the sequence. This problem is equivalent to approximating the support size of a distribution where every element in the support of the distribution has probability that is a multiple of $1/n$ [12]. Several works [12, 16, 15] gave close to linear lower bounds for distinguishing between support size at least n/d_1 and support size at most n/d_2 (for constant d_1 and d_2), where the best lower bound, due to Valiant and Valiant [15], is $\Omega(n/\log(n))$, and this bound is tight [15].

Turning to the upper bounds, assume first that we have a promise that the function f is a linear function, and we want to distinguish between the case that it depends on at most k variables and the case that it depends on more than $2k$ variables. Suppose we select a subset S of the variables by including each variable in the subset, independently, with probability $1/2k$. The first basic observation is that the probability that S contains at least one of the relevant variables of f when f depends on more than $2k$ variables, is some constant multiplicative factor (greater than 1) larger than the probability that this occurs when f depends on at most k relevant variables. The second observation is that given the promise that f is a linear function, using a small number of queries we can distinguish with high constant probability between the case that S contains at least one relevant variable of f, and the case that it contains no such variable. By quantifying the above more precisely, and repeating the aforementioned process a sufficient number of times, we can obtain Theorem 1.3

The algorithm for degree-d polynomials is essentially the same, except that the sub-test for determining whether S contains any relevant variables is more costly. The same ideas are also the basis for the algorithm for general functions, only we need a more careful analysis since in a general function we may have relevant variables that have very small influence. Indeed, as in previous work on testing k-juntas [7, 2, 3], the influence of variables (and subsets of variables), plays a central role (and we use some of the claims presented in previous work).

ORGANIZATION. We start by introducing several definitions and basic claims in Section 2. In Section 3 we prove Theorems 1.1 and 1.2 (the lower and upper bounds for general functions). In Section 4 we describe our results for restricted function classes, where the algorithms for linear functions and more generally, for degree-d polynomials, are special cases of a slight variant of the algorithm for general functions. Finally, in Section 5 we discuss extending the results to general finite domains and ranges, with arbitrary product distributions.

2 Preliminaries

For two functions $f, g : \{0,1\}^n \to \{0,1\}$, we define the *distance* between f and g as $\Pr_x[f(x) \neq g(x)]$ where x is selected uniformly in $\{0,1\}^n$. For a family of functions $\mathcal{F}$ and a function f, we define the distance between f and $\mathcal{F}$ as the minimum distance over all $g \in \mathcal{F}$ of the distance between f and g. We say that f is *ϵ-far* from $\mathcal{F}$, if this distance is at least ϵ.

Our work refers to the influence of sets of variables on the output of a Boolean function (in a way that will be described presently). As such, we often consider the values that a function f attains conditioned on a certain fixed assignment to some of its variables, e.g., the values f may

take when the variables x_1 and x_3 are set to 0. For an assignment σ to a set of variables S we will denote the resulting restricted function by $f_{S=\sigma}$. Thus, $f_{S=\sigma}$ is a function of $\{0,1\}^{n-|S|}$ variables. When we wish to relate to the variables $\{x_1,\ldots,x_n\}\setminus S$ we use the notation $\bar{S}$.

We now give a definition that is central for this work:

Definition 2.1 *For a function $f:\{0,1\}^n\to\{0,1\}$ we define the* influence *of a set of variables $S\subseteq\{x_1,\ldots,x_n\}$ as $\Pr_{\sigma,y,y'}[f_{\bar{S}=\sigma}(y)\neq f_{\bar{S}=\sigma}(y')]$ where σ is selected uniformly at random from $\{0,1\}^{n-|S|}$ and y,y' are selected uniformly at random from $\{0,1\}^{|S|}$. For a fixed function f we denote this value by $I(S)$. When the set S consists of a single variable x_i we may use the notation $I(x_i)$ instead of $I(\{x_i\})$.*

Proofs of the following claims can be found in [7]. The first claim tells us that the influence of sets of variables is monotone and subadditive:

Claim 2.1 *Let f be a function from $\{0,1\}^n$ to $\{0,1\}$, and let S and T be subsets of the variables $x_1,\ldots,x_n$. It holds that $I(S)\leq I(S\cup T)\leq I(S)+I(T)$.*

Definition 2.2 *For a fixed function f the* marginal influence *of set of variables T with respect to a set of variables S is $I(S\cup T)-I(S)$. We denote this value by $I^S(T)$.*

The marginal influence of a set of variables is diminishing:

Claim 2.2 *Let S, T, and W be disjoint sets of variables. For any fixed function f it holds that $I^S(T)\geq I^{S\cup W}(T)$.*

The next claim relates between the distance to being a k-junta and the influence of sets of variables.

Claim 2.3 *Let f be a function that is ϵ-far from being a k-junta. Then for every subset S of f's variables of size at most k, the influence of $\{x_1,\ldots,x_n\}\setminus S$ is at least ϵ.*

The converse of Claim 2.3 follows from the definition of influence:

Claim 2.4 *Let f be a function such that for every subset S of f's variables of size at most k, the influence of $\{x_1,\ldots,x_n\}\setminus S$ is at least ϵ. Then f is ϵ-far from being a k-junta.*

3 Distinguishing between k-Juntas and Functions Far From Every $(1+\gamma)k$-Junta

In this section we prove Theorems 1.1 and 1.2 (stated in the introduction).

3.1 The Lower Bound

The lower bound stated in Theorem 1.1 is achieved by a reduction from the *Distinct Elements* problem. In the Distinct Elements problem an algorithm is given query access to a string s and must compute approximately and with high probability the number of distinct elements contained in s. For a string of length t, this problem is equivalent to approximating the support size of a distribution where the probability for every event is in multiples of $1/t$ [12]. Valiant and Valiant [15] give the following theorem (paraphrased here):

Theorem 3.1 *For any constant $\varphi > 0$, there exists a pair of distributions p^+, p^- for which each domain element occurs with probability at least $1/t$, satisfying:*

1. $|S(p^+) - S(p^-)| = \varphi \cdot t$, *where* $S(D) \stackrel{\text{def}}{=} |\{x : \Pr_D[x] > 0\}|$.
2. *Any algorithm that distinguishes p^+ from p^- with probability at least $2/3$ must obtain $\Omega(\frac{t}{\log(t)})$ samples.*

While the construction in the proof of this theorem relates to distributions where the probability of events is not necessarily a multiple of $1/t$, it carries to the Distinct Elements problem [17].

In our work we use the following implication of this theorem - $\Omega(t/\log(t))$ queries are required to distinguish between a string of length t with $\frac{t}{2}$ distinct elements and one with fewer than $\frac{t}{16}$ distinct elements (for a sufficiently large t).[4]

In what follows we assume $k = n/8$, and later we explain how to (easily) modify the argument for the case that $k \leq n/8$ by "padding". We set $\gamma = 1$ (so that $1+\gamma = 2$), which implies the bound holds for all $\gamma \leq 1$. Using terminology coined by Raskhodnikova et al. [12], we refer to each distinct element in the string as a "color". We show a reduction that maps strings of length $t = \Theta(n)$ to functions from $\{0,1\}^n$ to $\{0,1\}$ such that the following holds: If there exists an algorithm that can distinguish (with high constant probability) between functions that are k-juntas and functions that are ϵ-far from any $2k$-junta (for a constant ϵ) using q queries, then the algorithm can be used to distinguish between strings with at most $k - \Theta(\log(k))$ colors and strings with at least $8k - \Theta(\log(k))$ colors using q queries.

We begin by describing a parametrized family of functions, which we denote by F_m^n. Each function in F_m^n depends on the first $\log(n)$ variables and on an additional subset of m variables.[5] The first $\log(n)$ variables are used to determine the identity of one of these m variables, and the value of the function is the assignment to this variable. More formally, for each subset $U \subset \{\log(n)+1, \ldots, n\}$ of size m and each surjective function $\psi : \{0,1\}^{\log(n)} \to U$, we have a function $f^{U,\psi}$ in F_m^n where $f^{U,\psi}(y_1, \ldots, y_n) = y_{\psi(y_1,\ldots,y_{\log(n)})}$. For a given function $f^{U,\psi}$ we call the variables $\{x_i\}_{i \in U}$ *active variables*.

Claim 3.1 *For any constant value c and for $t > n/c$, every function in $F_{t/2}^n$ is ϵ-far from all $t/4$-juntas, for a* constant *value ϵ.*

Proof: From Claim 2.4 we know that it suffices to show that for every function $f \in F_{t/2}^n$, and for every set of variables $S \subset \{x_1, \ldots, x_n\}$ having size at most $t/4$, the set of variables $\bar{S} = \{x_1, \ldots, x_n\} \setminus S$ has influence at least ϵ for a constant ϵ.

Consider a particular function $f \in F_{t/2}^n$. For any set S having size at most $t/4$, the set $\bar{S}$ contains at least $t/4$ active variables. We next show that the influence of a set T of $t/4$ active variables is at least $1/8c$, and by the monotonicity of the influence (Claim 2.1) we are done. The influence of T is defined as $\Pr_{\sigma,y,y'}(f_{\bar{T}=\sigma}(y) \neq f_{\bar{T}=\sigma}(y'))$ where σ is selected uniformly at random from $\{0,1\}^{n-|T|}$ and y, y' are selected uniformly at random from $\{0,1\}^{|T|}$. The probability of

[4] We note that allowing a bigger gap between the number of distinct elements (e.g., distinguishing between strings with at least t/d distinct elements for some constant d and strings with at most $t^{1-\alpha}$ distinct elements for a (small) constant α), does not make the distinguishing task much easier: $\Omega(t^{1-o(1)})$ queries are still necessary [12].

[5] In fact, it depends on an integer number of variables, and thus depends, e.g., on the $\lceil \log n \rceil$ first variables. We ignore this rounding issue throughout the paper, as it makes no difference asymptotically.

$x_{\psi(\sigma_1,\ldots,\sigma_{\log(n)})}$ belonging to T is at least $|T|/n = t/4 \geq n/4c$. The probability of this coordinate having different values in y and y' is $1/2$, and the claim follows. ■

We now introduce the reduction $R(s)$, which maps a string of colors s (a potential input to the distinct elements problem) to a function from $\{0,1\}^n$ to $\{0,1\}$ (a potential input to the "k-junta vs. far from $(1+\gamma)k$-junta" problem):

Let s be a string of length n, where every element i in s gets a color from the set $\{1,\ldots,n-\log(n)\}$, which we will denote by $s[i]$. The mapping $R(s) = f$ maps a string with m colors to a function in F_m^n. Informally, we map each color to one of the variables $x_{\log(n)+1},\ldots,x_n$ in f's input, and compute $f(y_1,\ldots,y_n)$ by returning the value of the variable that corresponds to the color of the element in s indexed by the values $y_1,\ldots,y_{\log(n)}$. More precisely, let $b : \{0,1\}^{\log(n)} \rightarrow \{0,\ldots,n-1\}$ be the function that maps the binary representation of a number to that number, e.g., $b(010) = 2$. We define the function f (that corresponds to a string s) as follows: $f(y_1,\ldots,y_n) = y_{s[b(y_1,\ldots,y_{\log(n)})]+\log(n)}$ (recall that the colors of s range from 1 to $n-\log(n)$).
The next claim follows directly from the definition of the reduction.

Claim 3.2 *The reduction $R(s)$ has the following properties:*

1. *For a string s, each query to the function $f = R(s)$ of the form $f(y_1,\ldots,y_n)$ can be answered by performing a single query to s.*
2. *For a string s with $n/2$ colors the function $f = R(s)$ belongs to $F_{n/2}^n$.*
3. *For a string s with $n/16$ colors the function $f = R(s)$ belongs to $F_{n/16}^n$.*

By Claims 3.1 and 3.2, any algorithm that can distinguish (with high constant probability) between functions that are $n/8$-juntas and functions that are ϵ-far from all $n/4$-juntas can be used to distinguish (with high constant probability) between strings with $n/2$ distinct elements and strings with $n/16$ distinct elements. Given the lower bound from [15], we have that any algorithm that distinguishes (with high constant probability) between functions with at most $n/8$ relevant variables and functions that are ϵ-far from all functions with at most $n/4$ relevant variables must perform $\Omega(n/\log(n))$ queries.

Dealing with general $k \leq n/8$: In the reduction R described above we have a number of relevant variables linear in n. We wish to show that we cannot distinguish in better time between k-Juntas and functions far from every $2k$-Junta when $k = o(n)$. This can be established by "padding" the function in the reduction as follows: Let the input to the reduction now be a string s of length $t = \Theta(k)$. The reduction in the setting described above maps such a string to a function f from $\{0,1\}^t$ to $\{0,1\}$. We can define a modified function f', which gets as input $y \in \{0,1\}^n$ and returns $f(y_1,\ldots,y_t)$.

Given the reduction above and the generalization to $k \leq n/8$ we obtain Theorem 1.1.

3.2 The Algorithm

In this subsection we present the algorithm referred to in Theorem 1.2. This algorithm uses the procedure **Test-for-relevant-variables** (given in Figure 1), which performs repetitions of the *independence test* defined in [7]. The number of repetitions depends on the parameters η and δ, which the algorithm receives as input.

Test-for-relevant-variables
Input: Oracle access to a function f, a set S of variables to examine, an influence parameter η and a confidence parameter δ.

1. Repeat the following $m = \Theta(\log(1/\delta)/\eta)$ times:
   (a) Select $\sigma \in \{0,1\}^{n-|S|}$ uniformly at random.
   (b) Select two values $y, y' \in \{0,1\}^{|S|}$ uniformly at random. If $f_{\bar{S}=\sigma}(y) \neq f_{\bar{S}=\sigma}(y')$ return true.
2. Return false.

Figure 1: Test-for-relevant-variables.

Claim 3.3 *When given access to a function f, a set S, and parameters η and δ, where S has influence of at least η,* **Test-for-relevant-variables** *returns* true *with probability at least $1-\delta$. When S contains no relevant variables,* **Test-for-relevant-variables** *returns* false *with probability 1. It performs $\Theta(\log(1/\delta)/\eta)$ queries.*

Claim 3.3 follows directly from the definition of influence and a standard amplification argument.

Separate-k-from-$(1+\gamma)k$
Input: Oracle access to a function f, an approximation parameter $\gamma < 1$ and a distance parameter ϵ.

1. Repeat the following $m = \Theta(1/\gamma^2)$ times:
   (a) Select a subset S of the variables, including each variable in S independently with probability $1/2k$.
   (b) Run **Test-for-relevant-variables** on f and S, with influence parameter $\eta = \Theta(\epsilon/k)$ and with confidence parameter $\delta = 1/8m$.
2. If the fraction of times that **Test-for-relevant-variables** returned true passes a threshold τ, return more-than-$(1+\gamma)k$. Otherwise return up-to-k. We determine τ in the analysis.

Figure 2: Separate-k-from-$(1+\gamma)k$.

Proof of Theorem 1.2: We prove that the statement in the theorem holds for Algorithm **Separate-k-from-$(1+\gamma)k$**, given in Figure 2. For a function f that has at most k relevant variables (i.e., is a k-junta), the probability that S (created in Step 1a of **Separate-k-from-$(1+\gamma)k$**) contains at least one such relevant variable is (at most) $p_k = 1 - (1 - \frac{1}{2k})^k$ (note that $1/4 < p_k \leq 1/2$). It follows from the one-sided error of **Test-for-relevant-variables** that the probability that it will return true in Step 1b is at most this p_k. We will show that if f is ϵ-far from every $(1+\gamma)k$-junta, then the probability of **Test-for-relevant-variables** returning true in Step 1b is at least $p'_k = p_k + \Omega(\gamma)$. Having established this, the correctness of **Separate-k-from-$(1+\gamma)k$** follows by setting the threshold τ to $\tau = (p_k + p'_k)/2$.

In the following we assume that when applied to a subset of the variables with influence at least η, **Test-for-relevant-variables** executed with the influence parameter η, returns true. We will later factor the probability of this not happening in even one iteration of the algorithm into our analysis of the algorithm's probability of success.

Consider a function f that is ϵ-far from every $(1+\gamma)k$-junta. For such a function, and for any constant $c > 1$, by Claim 2.3 one of the following must hold.

1. There are at least $(1+\gamma)k$ variables in f each with influence at least $\epsilon/c(1+\gamma)k$.
2. There are (more than $c(1+\gamma)k$) variables each with influence less than $\epsilon/c(1+\gamma)k$ that have, as a set, an influence of at least ϵ.

To verify this, note that if Case 1 does not hold, then there are fewer than $(1+\gamma)k$ variables in f with influence at least $\epsilon/c(1+\gamma)k$. Recall that by Claim 2.3, the variables of f except for the $(1+\gamma)k$ most influential variables have a total influence of at least ϵ, giving us Case 2.

We first deal with Case 1 (which is the simpler case). We wish to show that the probability that S (as selected in Step 1a) contains at least one variable with influence $\Omega(\epsilon/(1+\gamma)k)$ is $p_k + \Omega(\gamma)$. As there are at least $(1+\gamma)k$ variables with influence $\Omega(\epsilon/(1+\gamma)k)$, it suffices to consider the influence attributed to these variables, and to bound from below the probability that at least one of them appears in S. If we consider these $(1+\gamma)k$ variables one after the other (in an arbitrary order), for the first k variables, the probability that (at least) one of them is assigned to S is p_k (as defined above). If none of these were assigned to S, an event that occurs with probability at least $1 - p_k \geq 1/2$, we consider the additional γk variables. The probability of at least one of them being selected is at least γp_k, and so we have that the total probability of S containing at least one variable with influence $\Omega(\epsilon/(1+\gamma)k)$ is at least $p_k(1+\gamma/2)$. Given that $p_k > 1/4$ we have that the probability is at least $p_k + \gamma/8$, as required.

For our analysis of Case 2 we will focus on the set of variables described in the case. Recall that this set has influence of at least ϵ while every variable in the set has influence of less than $\epsilon/c(1+\gamma)k$. We denote this set of variables by $Y = \{y_1, \ldots, y_\ell\}$. We wish to bound from below the influence of subsets of Y. To this end we assign to each variable from the set Y a value that bounds from below the marginal influence it has when added to any subset of Y. By the premise of the claim we have that $I(Y) \geq \epsilon$. We consider the values $I(y_1), I^{\{y_1\}}(y_2), \ldots, I^{\{y_1,\ldots,y_{\ell-1}\}}(y_\ell)$. The sum of these must be at least ϵ by the definition of marginal influence (Definition 2.2). Let us denote by $I'(y_i)$ the value $I^{\{y_1,\ldots,y_{i-1}\}}(y_i)$. We refer to this as *the* marginal influence of y_i. If we consider adding (with probability $1/2k$) each element in Y to S in the order $y_1, \ldots, y_\ell$, we get by Claim 2.2 that the total influence of S is no less than the total of the marginal influences of those variables added to S. It now suffices to show that the sum of marginal influences in S is likely to be at least $\epsilon/4k$, and we are done.

To see that the sum of marginal influences in S is likely to be $\Omega(\epsilon/k)$, we first define the random variables $\{\chi_i\}$. The variable χ_i gets the value of $\frac{c(1+\gamma)k}{\epsilon} I'(y_i)$ if y_i is selected and 0 otherwise. We have:

$$\mathrm{Exp}[\chi_i] = \frac{1}{2k}\frac{c(1+\gamma)k}{\epsilon} I'(y_i) = \frac{c(1+\gamma)}{2\epsilon} I'(y_i)\ . \tag{1}$$

By the linearity of expectation we have

$$\mathrm{Exp}\left[\sum_{i=1}^{\ell} \chi_i\right] = \sum_{i=1}^{\ell} \mathrm{Exp}[\chi_i] = \frac{c(1+\gamma)}{2\epsilon} \sum_{i=1}^{\ell} I'(y_i) \geq \frac{c}{2}\ . \tag{2}$$

Using a multiplicative form of the Chernoff bound we know that

$$\Pr\left[\sum_{i=1}^{\ell}\chi_i < \frac{1}{2}\mathrm{Exp}\left[\sum_{i=1}^{\ell}\chi_i\right]\right] \leq e^{-\frac{c}{16}} \ . \tag{3}$$

For an appropriately selected c this means we are unlikely to have $\sum_{i=1}^{\ell}\chi_i$ that is less than a constant[6], and therefore we are likely to have

$$\sum_{y_i\in S} I'(y_i) = \frac{\epsilon}{c(1+\gamma)k}\sum_{i=1}^{\ell}\chi_i = \Omega(\epsilon/k) \ , \tag{4}$$

as required.

We now turn to lower bounding the algorithm's probability of success. By the choice of $\delta = 1/8m$, the probability that *any* of the m runs of **Test-for-relevant-variables** fails to detect a set with influence $\Omega(\epsilon/(1+\gamma)k)$ is at most $1/8$. Conversely, when the set S contains no variables with influence, **Test-for-relevant-variables** never accepts. Thus, for a function with at most k relevant variables, **Test-for-relevant-variables** accepts with probability at most p_k. On the other hand, for a function that is ϵ-far from all functions with at most $(1+\gamma)k$ relevant variables,**Test-for-relevant-variables** accepts with probability at least $p_k + \gamma/8$. We therefore set the threshold τ to $p_k + \gamma/16$. Recall that the number of iterations performed by the algorithm is $m = \Theta(1/\gamma^2)$. By an additive Chernoff bounds (for a sufficiently large constant in the Θ notation), conditioned on **Test-for-relevant-variables** returning a correct answer in each iteration, the probability that we "fall on the wrong side of the threshold" is at most $1/8$. Try to improve phrasing. Thus, with probability at least $3/4$ our algorithm returns a correct answer.

Finally, we bound the query complexity of the algorithm. The algorithm perform $m = \Theta(1/\gamma^2)$ iterations. In each iteration it runs **Test-for-relevant-variables** with influence parameter $\eta = \Theta(\epsilon/k)$ and with confidence parameter $\delta = 1/8m$. The query complexity of the procedure **Test-for-relevant-variables** is $\Theta(\log(1/\delta)/\eta)$, giving a total of $\Theta(\frac{k\log(1/\gamma^2)}{\gamma^2\epsilon})$ queries. ■

4 Restricting the Problem to Classes of Functions

Given that in general, distinguishing between functions that are k-juntas and functions that are ϵ-far from $(1+\gamma)k$ juntas requires an almost linear dependence on k, we ask whether this task can be performed more efficiently for restricted function classes (and possibly without the introduction of the distance parameter ϵ). In particular, let $\mathcal{C}_\eta$ be the class of functions where every variable has influence at least η. As we shall see later, there are natural families of functions that are subclasses of $\mathcal{C}_\eta$.

Theorem 4.1 *Given query access to a function $f \in \mathcal{C}_\eta$, it is possible to distinguish with high constant probability between the case that f has at most k relevant variables and the case that f has more than $(1+\gamma)k$ relevant variables by performing $\Theta(\frac{\log(1/\gamma)}{\gamma^2\eta})$ queries.*

Proof: We use the exact same algorithm as we use in the general case (that is, **Separate-k-from-$(1+\gamma)k$** given in Figure 2) with the following exception. In Step 1b, instead of setting the influence parameter to $\Theta(\epsilon/k)$, we set it to $\Theta(\eta)$. The proof of correctness follows Case 1 in the general proof of correctness. ■

[6]Recall that we can select c. It determines the constant hidden in the algorithms Θ notation for ϵ'.

4.1 Linear Functions

A well studied class of functions for which we can test whether a function in the class has k relevant variables or more than $(1+\gamma)k$ relevant variables, by performing a number of queries that depends only on γ, is the class of linear functions. For each function in the class, every influential variable has influence $1/2$. As a corollary of Theorem 4.1 we get Theorem 1.3 (stated in the introduction).

A natural question is whether this result can be improved to distinguish between, e.g., linear functions that depend on at most k variables and linear functions that depends on more than k variables. While distinguishing between linear functions that depend on k vs. $k+1$ variables is easy (simply compare $f(\vec{0})$ to $f(\vec{1})$), Goldreich [8] presents two families of linear functions, one with $n/2$ relevant variables and one with $n/2+2$ variables, and shows they can't be distinguished with $o(\sqrt{n})$ queries. Building on another result of Goldreich [8], Chakraborty et al. [5] show that it is not possible to distinguish with constant success probability between linear functions with at most k variables and linear functions with at least $k+2$ variables by performing $o(k/\text{polylog}(k))$ queries. Finally, Blais et al. [4] show that $\Omega(\min(k, n-k))$ queries are required to distinguish between such functions.

4.2 Polynomials over $GF(2)$

It is well known that every Boolean function can be represented by a polynomial over $GF(2)$. Such a polynomial is the parity of several monomials. That is, a function f can be written as $\bigoplus_i \phi^i$ where every monomial ϕ^i is the product of variables, i.e., $\phi^i = \Pi_{j\in J_i} x_j$ where $J_i \subseteq [n]$. Monomials over $GF(2)$ have a natural logical interpretation, and from here on we think of monomials as conjunctions of variables, that is, $\phi^i = \bigwedge_{j\in J_i} x_j$ where $J_i \subseteq [n]$. The degree of a polynomial p is the number of variables in the largest monomial in p. It is convenient for us to work with a small variation on the concept of monomials.

Definition 4.1 *A* Generalized Monomial *over $GF(2)$ is a conjunction of literals (variables* and their negations*).*

We note that if a function f can be computed as the parity of generalized monomials with a number of variables at most d in each such generalized monomial, it can also be computed by a "standard" polynomial with degree at most d. As polynomials in this section are characterized by their degree, we describe them without loss of generality as the parity of generalized monomials.

We first wish to show (using Theorem 4.1) that we can distinguish between polynomials of degree at most d with at most k variables and those with at least $(1+\gamma)k$ variables using $O(2^d \log(1/\gamma)/\gamma^2)$ queries. We will then show that the exponential dependence on d cannot be significantly improved.

The following is a well known fact:

Claim 4.1 *Let us denote by P_h the probability of a function h to take the value* 1 *when the input is chosen uniformly at random, and let p be a polynomial of degree d that isn't the* 0 *polynomial. It holds that $P_p \geq 2^{-d}$.*

The proof follows by induction on d. We include the proof of the following claim for the sake of completeness:

Claim 4.2 *Let p be a polynomial of degree d. For every variable x_j in p such that $I(x_j) \neq 0$ it holds that $I(x_j) \geq 2^{-d}$.*

Proof: Let $p = \sum_{i=1}^{m} \phi^i$ be a polynomial of degree d. We consider, without loss of generality, the influence of the variable x_1 that appears in the monomials $\phi^1, \ldots, \phi^k$. The variable x_1 effects the value of p (given an assignment to all other variables) when the polynomial $p' = \sum_{i=1}^{k} \phi^i_{x_1=1}$ does not equal 0. Indeed, the influence of x_1 is exactly half the probability that p' does not equal 0. As p' is of degree at most $d-1$ this happens with probability at least 2^{-d+1} by Claim 4.1, and thus $I(x_j) \geq 2^{-d}$ as required. ■

The next theorem now follows from Claim 4.2 and Theorem 4.1:

As in the proof of Theorem 1.1 we perform a reduction from the Distinct Elements problem. We now describe a parametrized family of functions, which we denote $F^n_{m,d}$.[7] Each function in $F^n_{m,d} : \{0,1\}^n \to \{0,1\}$ is a polynomial of degree d that depends on the first $d-1$ variables and on an additional subset of m variables. The setting of the first $d-1$ variables determines a particular subset of the m variables, of size $r = (n-d+1)/2^{d-1}$, and the value of f is the parity of the variables in this subset. More formally, let the sets $U^1, \ldots, U^{2^{d-1}}$ be consecutive sets of variables from the variables $x_d, \ldots, x_n$. That is, $U^1 = \{x_d, \ldots, x_{d+r-1}\}, U^2 = \{x_{d+r}, \ldots, x_{d+2r-1}\}$ etc. Let $\Psi : \{0,1\}^{d-1} \to \{1, \ldots, 2^{d-1}\}$ be a function that maps an assignment of the first $d-1$ variables to m/r values in the range $\{1, \ldots, 2^{d-1}\}$. All functions of the form $f(x_1, \ldots, x_n) = \bigoplus U^{\Psi(x_1,\ldots,x_{d-1})}$ (and only these functions) are members of $F^n_{m,d}$, where $\bigoplus U$ is used to denote the parity of all variables in a set U. We refer to variables in $\{x_d, \ldots, x_n\}$ that are relevant variables as *active variables*. Observe that the total number of relevant variables for each function in $F^n_{m,d}$ is $m+d-1$. Here we consider $m = \Theta(n)$, so that the number of relevant variables in $\Theta(n)$ as well. As in the case of the lower bound for general functions, the argument can be easily adapted to a number of relevant variables that is significantly smaller than n using "padding".

Claim 4.3 *Each function in $F^n_{m,d}$ is realizable by a degree-d polynomial.*

Proof: To prove the claim consider a polynomial that has, for every assignment $y_1, \ldots, y_{d-1}$ to the first $d-1$ variables, and for the set U that corresponds to it, $|U|$ generalized monomials. Each of these generalized monomial has d literals - a variable in U and for each $1 \leq i \leq d-1$, the literal x_i if $y_i = 1$, and the literal $\bar{x}_i$ if $y_i = 0$. Such a polynomial is of degree d (as all generalized monomials in it are over d literals) and computes a function in $F^n_{m,d}$ (since for the assignment $y_1, \ldots, y_{d-1}$, by the definition of polynomials, the function takes the value $\bigoplus U$). Furthermore, such a polynomial exists for every function in $F^n_{m,d}$. ■

Claim 4.4 *Functions in $F^n_{n/2,d}$ are ϵ-far from all functions with at most $n/4$ relevant variables, for a* constant *value ϵ.*

Proof: From Claim 2.4 we know that it suffices to show that for every function $f \in F^n_{n/2,d}$, and for every subset $S \subset \{x_1, \ldots, x_n\}$ of size at most $n/4$, the set $\bar{S} = \{x_1, \ldots, x_n\} \setminus S$ has influence at least ϵ.

Consider a particular function $f \in F^n_{n/2,d}$. For any set S of size at most $n/4$ the set $\bar{S}$ contains more than $n/8$ active variables (for a sufficiently large n). These variables must belong to at least $\frac{n/8}{r} > \frac{n/8}{n/2^{d-1}} = \frac{2^{d-1}}{8}$ different sets $\{U^i\}$. As these are active variables, each such set U^i has at least one assignment $x_1, \ldots, x_{d-1} = y_1, \ldots, y_{d-1}$ such that $f_{\{x_1,\ldots,x_{d-1}\}=y_1,\ldots,y_{d-1}} = \bigoplus U^i$. Let us denote the set of such assignments Y. That is,

$$Y = \{y_1, \ldots, y_{d-1} : \; U^{\psi(y_1,\ldots,y_{d-1})} \cap \bar{S} \neq \emptyset\} \; .$$

[7] We assume that m is a multiple of $(n-d+1)/2^{d-1}$.

In such a restricted function $f_{\{x_1,\ldots,x_{d-1}\}=y_1,\ldots,y_{d-1}}$ the set $\bar{S}$ has influence 1/2. Therefore we have that

$$I(\bar{S}) \geq \frac{1}{2}\mathrm{Pr}_{y\in\{0,1\}^n}[y_1 \ldots y_{d-1} \in Y] \geq \frac{1}{2}\frac{2^{d-1}}{8}/2^{d-1} = \frac{1}{16} ,$$

as required. ■

We now introduce the reduction $R(s)$, which maps a string of colors (a potential input to the distinct elements problem) to a degree-d polynomial from $\{0,1\}^n$ to $\{0,1\}$ (a potential input to the "*k vs. $(1+\gamma)k$-junta problem*" for degree-d polynomials):

Let s be a string of length 2^{d-1}, where every element i in s gets a color from the set $\{1,\ldots,2^{d-1}\}$, which we will denote by $s[i]$. We denote the number of distinct colors in s as $\chi(s)$. For a fixed value n the mapping $R(s) = f$ maps s to a function in $F^n_{\chi(s)r,d}$. We map each color to one of the sets $U^1,\ldots,U^{2^{d-1}}$ in f's input, and compute f's output on an input $y \in \{0,1\}^n$ by returning the parity of the input variables that correspond to the color of the element in s indexed by the values $y_1,\ldots,y_{d-1}$. More precisely, let $b : \{0,1\}^{d-1} \to \{0,\ldots,2^{d-1}-1\}$ be the function that maps the binary representation of a number to that number, e.g., $b(010) = 2$. We define the function f that corresponds to a string s as follows: $f(y_1,\ldots,y_n) = \bigoplus U^{s[b(y_1,\ldots,y_{d-1})+1]}$.

The next claim follows directly from the definition of the reduction:

Claim 4.5 *The reduction $R(s)$ has the following properties:*

1. *For a string s, each query to the function $f = R(s)$ of the form $f(y_1,\ldots,y_n)$ can be answered by performing a single query to s.*
2. *For a string s with $2^{d-1}/2$ colors, the function $f = R(s)$ belongs to $F^n_{(n-d+1)/2,d}$.*
3. *For a string s with $2^{d-1}/16$ colors, the function $f = R(s)$ belongs to $F^n_{(n-d+1)/16,d}$.*

As in the general case (and using Claims 4.3 and 4.4), this means that any algorithm that can distinguish (with high constant probability) between degree-d polynomials with at most $n/8$ relevant variables and degree-d polynomials that are ϵ-far from all degree-d polynomials with at least $n/4$ relevant variables can be used to distinguish strings of length 2^{d-1} that either have at most $2^{d-1}/16$ distinct elements or that have at least $2^{d-1}/2$ distinct elements. Given the lower bound from [15] we have that any algorithm that distinguishes (with high constant probability) degree-d polynomials with at most $n/8$ relevant variables from those that are ϵ-far from all degree-d polynomials with at least $n/4$ relevant variables must perform $\Theta(2^d/d)$ queries. Theorem 1.5 (which is stated for general k) follows by applying a "padding" argument as in the general case.

4.3 Monotone Functions

In this subsection we give a lower bound for the number of queries required to determine whether a monotone function depends on at most k variables or is ϵ-far from every function that depends on $2k$ variables. Here monotone functions are defined in the standard manner - we say a function $f : \{0,1\}^n \to \{0,1\}$ is monotone if for all $y, y' \in \{0,1\}^n$ it holds that $y > y' \Rightarrow f(y') > f(y')$. The relation $y > y'$ holds when $y_i \geq y'_i$ for all i, and $y_i > y'_i$ for some i. One could hope that restricting the family of functions we're dealing with to monotone functions could significantly decrease the number of required queries. This is the case for at least one property of Boolean functions - average influence [**?**]. We show:

Theorem 4.2 *Any algorithm that distinguishes (with constant probability) between monotone functions with k variables and monotone functions that are $\Theta(1/\sqrt{\log(k)})$-far from all those with $2k$ variables must perform $\Omega(k/\log(k))$ queries.*

It follows from Theorem 4.2 that any algorithm for the problem (stated in the claim) whose dependence on $1/\epsilon$ is polynomial, must perform a number of queries that is almost linear in k.

The construction for monotone functions is similar to that for general functions. The constructions differ in one aspect, leading the lower bound (for monotone functions) to hold only for algorithms that can distinguish between monotone functions that depend on k variables and functions that are $\Theta(1/\sqrt{\log(k)})$-far from those depending on $(1+\gamma)k$ variables.

Due to this similarity we only state the points where it differs from the general construction. We again describe a parametrized family of functions, which we denote M_m^n. Each function in M_m^n is monotone, and depends on the first $\log(n)$ variables and on an additional subset of m variables. For a function $f \in M_m^n$ and a value $y \in \{0,1\}^n$, if $\sum_{i=1}^{\log(n)} y_i < \lfloor \log(n)/2 \rfloor$ we have $f(y) = 0$. Likewise, if $\sum_{i=1}^{\log(n)} y_i > \lfloor \log(n)/2 \rfloor$ we have $f(y) = 1$. When we have exactly $\sum_{i=1}^{\log(n)} y_i = \lfloor \log(n)/2 \rfloor$, then the first $\log(n)$ variables are used to determine the identity of one of the m additional variables, and the value of the function is the assignment to this variable. More specifically, denoting by $\{0,1\}_{1/2}^{\ell}$ bit strings of length ℓ that contain exactly $\lfloor \ell/2 \rfloor$ values of 1, for each subset $U \subset \{\log(n)+1, \ldots, n\}$ of size m and each surjective function $\psi : \{0,1\}_{1/2}^{\log(n)} \to U$ we have a function $f^{U,\psi}$ in M_m^n where $f^{U,\psi}(y_1, \ldots, y_n) = y_{\psi(y_1,\ldots,y_{\log(n)})}$. For a given function $f^{U,\psi}$ we call the variables in the set U *active variables*.

The next claim follows directly from the definition of M_m^n.

Claim 4.6 *Functions in M_m^n are monotone.*

Claim 4.7 *Functions in $M_{n/2}^n$ are $\Theta(1/\sqrt{\log(n)})$-far from all $n/4$-juntas.*

To see that Claim 4.7 holds, observe that the distance of interest is only on assignments $y \in \{0,1\}^n$ where $\sum_{i=1}^{\log(n)} y_i = \lfloor \log(n)/2 \rfloor$. These constitute $\Theta(1/\sqrt{\log(n)})$ of all assignments. Claim 4.7 follows from an analysis similar to that of Claim 3.1.

The reduction from the Distinct Elements problem follows the same lines as the general case, with obvious modifications - the string we reduce from is of length $\Theta(n/\sqrt{\log(n)})$, and the positive and negative families of functions (as stated above) are $\Theta(1/\sqrt{\log(n)})$-far from each other. The proof of Theorem 4.2 follows lines similar to those used in the general case.

5 Extending the results to general finite domains and ranges

In this section we show that Theorem 1.2 extends to the more general case of functions over finite domains and ranges over product distributions, and that the same holds for Theorem 4.1. We also observe that Theorems 1.3 and 1.4 extend to linear functions and degree-d polynomials over finite fields, respectively.

5.1 Preliminaries

Let $f : Y \to R$ where $Y = Y_1 \times \cdots \times Y_n$ is a finite domain and R is a finite range. An input $y_1, \ldots, y_n \in Y$ to the function f is drawn according to a product distribution $D = D_1 \times D_2 \times \cdots \times D_n$. We assume that we can draw an input $y \in Y$ according to D (though it is not assumed that D is known). A case of special interesting, which we have dealt with up till now, is when $Y_i = \{0, 1\}$ for each i, $R = \{0, 1\}$, and D is the uniform distribution over $Y = \{0, 1\}^n$.

Given the underlying distribution D, for two functions $f, g : Y \to R$, we define the *distance* between f and g (with respect to D) as $\Pr_x[f(x) \neq g(x)]$ where x is selected from Y according to D. For a family of functions $\mathcal{F}$ and a function f, we define the distance between f and $\mathcal{F}$ as the minimum distance over all $g \in F$ of the distance between f and g (with respect to D). We say that f is *ϵ-far* from $\mathcal{F}$ (with respect to D), if this distance is greater or equal to ϵ.

We next extend the notion of the influence of a set of variables where we shall use the following notation: For a set S of variables, we let Y_S be the domain restricted to S (i.e., for $S = \{x_{i_1}, \ldots, x_{i_\ell}\}$ we have $Y_S = Y_{i_1} \times \ldots \times Y_{i_\ell}$), and let D_S be the product distribution induced on the variables in the set S.

Definition 5.1 *For a function $f : Y \to R$ we define the* influence *of a set of variables $S \subseteq \{x_1, \ldots, x_n\}$ as $\Pr_{\sigma,y,y'}[f_{\bar{S}=\sigma}(y) \neq f_{\bar{S}=\sigma}(y')]$ where σ is selected from $Y_{\bar{S}}$ according to $D_{\bar{S}}$ and y and y' are selected from Y_S according to D_S. For a fixed function f and distribution D we denote this value by $I(S)$. When the set S consists of a single variable x_i we may use the notation $I(x_i)$ instead of $I(\{x_i\})$.*

While Fischer et al. [7] address in some of their claims the case of general domains and ranges, they consider the notion of the *variation* of a set rather than the influence (as in Definition 5.1). When the function is a Boolean function, the two notions essentially coincide, but this is not the case for a larger range. However, Blais [3] considers the notion of the influence of a set, and hence we build on the claims that he establishes (and in one case provide a proof that we have not found elsewhere). In particular, Claim 2.1 extends to the general case of finite domains and ranges [3]:

Claim 5.1 *Let $f : Y \to R$ be a function and let S and T be subsets of the variables $x_1, \ldots, x_n$. It holds that $I(S) \leq I(S \cup T) \leq I(S) + I(T)$.*

The same holds for Claim 2.3 (whose proof can also be found in [3]), and the simple proof of Claim 2.4 is easily extended. We restate them here for the sake of completeness.

Claim 2.3. *Let f be a function that is ϵ-far from being a k-junta. Then for every subset S of f's variables of size at most k, the influence of $\{x_1, \ldots, x_n\} \setminus S$ is at least ϵ.*

Claim 2.4. *Let f be a function such that for every subset S of f's variables of size at most k, the influence of $\{x_1, \ldots, x_n\} \setminus S$ is at least ϵ. Then f is ϵ-far from being a k junta.*

The definition of the *marginal influence* of a set of variables (Definition 2.2) remains as is: $I^S(T) \stackrel{\text{def}}{=} I(S \cup T) - I(S)$ (for the extended notion of the influence). It only remains to prove Claim 2.2 for the general case.

Claim 2.2. *Let S, T, and W be disjoint sets of variables. For any fixed function $f : Y \to R$ it holds that $I^S(T) \geq I^{S \cup W}(T)$.*

Proof: For a set S of variables and an assignment σ to S from Y_S, we let $p_S(\sigma)$ be the probability that σ is selected according to the underlying distribution D_S. observe that:

$$\begin{aligned} I(T) &= \Pr_{\sigma,y,y'}[f_{\bar{T}=\sigma}(y) \neq f_{\bar{T}=\sigma}(y')] \\ &= 1 - \sum_{\sigma} p_{\bar{T}}(\sigma) \cdot \Pr_{y,y'}[f_{\bar{T}=\sigma}(y) = f_{\bar{T}=\sigma}(y')] \\ &= 1 - \sum_{\sigma} p_{\bar{T}}(\sigma) \cdot \sum_{\rho \in R} \Pr_{y,y'}[f_{\bar{T}=\sigma}(y) = \rho \text{ and } f_{\bar{T}=\sigma}(y') = \rho] \end{aligned}$$

(where $\sigma \in Y_{\bar{T}}$ is selected according to $D_{\bar{T}}$ and $y, y' \in Y_T$ are selected according to D_T). We would like to show that

$$I(S \cup T) - I(S) \geq I(S \cup W \cup T) - I(S \cup W) \; .$$

Let $Q = \overline{S \cup W \cup T}$. We introduce one more notation: For $\sigma \in Y_Q$, $\alpha \in Y_T$, $\beta \in Y_W$ and an output value $\rho \in R$, let $p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho)$ denote the probability that the output of the function f is ρ, conditioned on $Q = \sigma$, $T = \alpha$, and $W = \beta$, where the probability is taken over all assignments to the variables in S. Using this notation we have:

$$I(S \cup W \cup T) = 1 - \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\rho \in R} \left(\sum_{\alpha \in Y_T} \sum_{\beta \in Y_W} p_T(\alpha) p_W(\beta) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 ,$$

$$I(S \cup T\}) = 1 - \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\beta \in Y_W} p_W(\beta) \sum_{\rho \in R} \left(\sum_{\alpha \in Y_T} p_T(\alpha) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 ,$$

$$I(S \cup W\}) = 1 - \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\alpha \in Y_T} p_T(\alpha) \sum_{\rho \in R} \left(\sum_{\beta \in Y_W} p_W(\beta) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 ,$$

$$I(S) = 1 - \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\alpha \in Y_T} p_T(\alpha) \sum_{\beta \in Y_W} p_W(\beta) \sum_{\rho \in R} (p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho))^2 \; .$$

Therefore,

$$\begin{aligned} & I(S \cup T) - I(S) \\ &= \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\rho \in R} \\ &\qquad \sum_{\beta \in Y_W} p_W(\beta) \left(\sum_{\alpha \in Y_T} p_T(\alpha) (p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho))^2 - \left(\sum_{\alpha \in Y_T} p_T(\alpha) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 \right) \; . \end{aligned}$$

Similarly,

$$\begin{aligned} & I(S \cup W \cup T) - I(S \cup W) \\ &= \sum_{\sigma \in Y_Q} p_Q(\sigma) \sum_{\rho \in R} \\ &\qquad \left(\sum_{\alpha \in Y_T} p_T(\alpha) \left(\sum_{\beta \in Y_W} p_W(\beta) p_T(\alpha) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 \right. \\ &\qquad\qquad \left. - \left(\sum_{\alpha \in Y_T} p_T(\alpha) \sum_{\beta \in Y_W} p_W(\beta) p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho) \right)^2 \right) . \end{aligned}$$

Fixing σ and ρ, let us simplify our notations as follows. Let $|Y_T| = N$ and $|Y_W| = M$. For an arbitrary order over Y_T, let $a_r = p_T(\alpha)$ for α that is the rth element in Y_T, and similarly define $b_q = p_W(\beta)$ and $c_{r,q} = p^{\sigma,\alpha,\beta}_{Q,T,W}(\rho)$. We would like to show the following:

$$\begin{aligned} & \sum_{q=1}^{M} b_q \left(\sum_{r=1}^{N} a_r (c_{r,q})^2 - \left(\sum_{r=1}^{N} a_r c_{r,q} \right)^2 \right) \\ &\quad \geq \sum_{r=1}^{N} a_r \left(\sum_{q=1}^{M} b_q c_{r,q} \right)^2 - \left(\sum_{r=1}^{N} a_r \sum_{q=1}^{M} b_q c_{r,q} \right)^2 . \end{aligned}$$

Let us denote:

$$\Psi_{a_1,\ldots,a_N}(z_1,\ldots,z_N) = \sum_{r=1}^{N} a_r (z_r)^2 - \left(\sum_{r=1}^{N} a_r z_r \right)^2$$

Then we would like to show that:

$$\sum_{q=1}^{M} b_q \cdot \Psi_{a_1,\ldots,a_N}(c_{1,q},\ldots,c_{N,q}) \geq \Psi_{a_1,\ldots,a_N} \left(\sum_{q=1}^{M} b_q c_{1,q}, \ldots, \sum_{q=1}^{M} b_q c_{N,q} \right) \tag{5}$$

(where we may use $\sum_{r=1}^{N} a_r = 1$ and $\sum_{q=1}^{M} b_q = 1$). We next show that $\Psi = \Psi_{a_1,\ldots,a_N}$ is convex, and hence Equation (5) follows by Jensen's inequality.

In order to show that Ψ is convex, we consider the (Hessian) matrix H defined by $H_{i,j} = \frac{\partial^2 \Psi(z_1,\ldots,z_N)}{\partial z_i \partial z_j}$. We shall verify that H is positive semi-definite. We have that $H_{i,i} = 2(a_i - a_i^2)$, and $H_{i,j} = -2a_i a_j$ for $j \neq i$. In order to establish that H is positive semidefinite, we consider any vector $\vec{y} = y_1,\ldots,y_N$, and show that $\vec{y} H \vec{y}^t \geq 0$. We start by computing $\vec{w} = \vec{y} H$. Observe that the jth column of H, denoted H^j, is of the following form: $H^j_j = 2a_j - 2a_j^2$ and $H^j_i = -2a_i a_j$ for $i \neq j$. Therefore,

$$w_j = \vec{y} H^j = 2y_j a_j - 2y_j a_j^2 - \sum_{i \neq j} 2y_i a_i a_j = 2a_j y_j - 2a_j \sum_{i=1}^{n} y_i a_i .$$

Now,

$$\begin{aligned}\vec{y}H\vec{y}^t &= \sum_{j=1}^{n} w_j y_j \\ &= 2\sum_{j=1}^{N} a_j y_j^2 - 2\sum_{j=1}^{N}\Big(a_j y_j \cdot \sum_{i=1}^{N} y_i a_i\Big) \\ &= 2\left(\sum_{j=1}^{N} a_j y_j^2 - \Big(\sum_{j=N}^{n} a_j y_j\Big)^2\right) .\end{aligned}$$

Since $\sum_{i=1}^{N} a_i = 1$, by Jensen's inequality we get a non-negative value. ■

5.2 Extending Theorem 1.2

We claim that Theorem 1.2 extends to general finite domains and ranges.

Theorem 5.1 *There exists an algorithm that, given query access to $f : Y \to R$, sampling access to a product distribution D over Y, and parameters $k \geq 1$, and $0 < \epsilon, \gamma < 1$, distinguishes with high constant probability between the case that f is a k-junta and the case that f is ϵ-far from any $(1+\gamma)k$-junta. The algorithm performs $O\left(\frac{k\log(1/\gamma)}{\epsilon\gamma^2}\right)$ queries.*

The algorithm referred to in Theorem 5.1 is Algorithm **Separate-k-from-$(1+\gamma)k$**, which remains exactly as is. Algorithm **Test-for-relevant-variables**, which is called as a subroutine from Algorithm **Separate-k-from-$(1+\gamma)k$** remains as is except that σ is selected from $Y_{\bar{S}}$ according to $D_{\bar{S}}$, and y and y' are selected from Y_S according to D_S. The proof of Theorem 5.1 is the same as the proof of Theorem 1.2 (where it relies on Claim 2.3 and Claim 2.2, which holds for general functions over finite domains and ranges). Theorem 4.1 is established as before.

5.3 Extending Theorems 1.3 and 1.4

Let F be a finite field. Here we consider the case that $Y = F^n$, $R = F$, and D is the uniform distribution over F^n. For every linear function $f : F^n \to F$ we have that each relevant variable has influence $1 - \frac{1}{|F|}$ (where influence is measured with respect to the uniform distribution). As a corollary of Theorem 4.1 we get:

Theorem 5.2 *Given query access to a linear function $f : F^n \to F$ (with the uniform distribution on inputs), it is possible to distinguish with high constant probability between the case that f has at most k relevant variables and the case that f has more than $(1+\gamma)k$ relevant variables by performing $\Theta(\log(1/\gamma)/\gamma^2)$ queries.*

Now consider a polynomial $f : F^n \to F$ of degree d. The probability such a polynomial takes the value 0 is at most $\left(\frac{|F|-1}{|F|}\right)^d$, and thus, similarly to what was proved in Claim 4.2, every variable in such a polynomial has influence at least $\left(\frac{|F|-1}{|F|}\right)^d$. As a corollary of Theorem 4.1 we get:

Theorem 5.3 *Given query access to a polynomial $f : F^n \to F$ of degree d (with the uniform distribution on inputs), it is possible to distinguish with high constant probability between the case that f has at most k relevant variables and the case that f has more than $(1+\gamma)k$ relevant variables by performing $O\left(\frac{|F|^d}{(|F|-1)^d} \cdot \frac{\log(1/\gamma)}{\gamma^2}\right)$ queries.*

References

[1] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. *SIAM Journal on Discrete Math*, 16(3):393–417, 2003.

[2] E. Blais. Improved bounds for testing juntas. In *Proceedings of the Twelveth International Workshop on Randomization and Computation (RANDOM)*, pages 317–330, 2008.

[3] E. Blais. Testing juntas nearly optimally. In *Proceedings of the Fourty-First Annual ACM Symposium on the Theory of Computing*, pages 151–158, 2009.

[4] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. To appear in the 26th Conference on Computational Complexity (CCC), 2011.

[5] S. Chakradorty, D. García-Soriano, and A. Matsliah. Private communication, 2010.

[6] H. Chockler and D. Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.

[7] E. Fischer, G. Kindler, D. Ron, S. Safra, and S. Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004.

[8] O. Goldreich. On testing computability by small width OBDDs. In *Proceedings of the Fourteenth International Workshop on Randomization and Computation (RANDOM)*, pages 574–587, 2010.

[9] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[10] M. Kearns and D. Ron. Testing problems with sub-learning sample complexity. *Journal of Computer and System Sciences*, 61(3):428–456, 2000.

[11] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2):165–183, 2002.

[12] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bonds for approximating distributions support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.

[13] D. Ron and G. Tsur. On approximating the number of relevant variables in a function. Technical Report TR11-041, Electronic Colloquium on Computational Complexity (ECCC), 2011.

[14] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[15] G. Valiant and P. Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of the Fourty-Third Annual ACM Symposium on the Theory of Computing*, pages 685–694, 2011. See also ECCC TR10-179 and TR10-180.

[16] P. Valiant. Testing symmetric properties of distributions. In *Proceedings of the Fourtieth Annual ACM Symposium on the Theory of Computing*, pages 383–392, 2008.

[17] P. Valiant. Private communications, 2011.