Is your function low-dimensional?

Anindya De ANINDYAD@SEAS.UPENN.EDU
University of Pennsylvania

Elchanan Mossel ELMOS@MIT.EDU
Massachusetts Institute of Technology

Joe Neeman JNEEMAN@MATH.UTEXAS.EDU
University of Texas, Austin

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

We study the problem of testing if a function depends on a small number of linear directions of its input data. We call a function f a *linear k-junta* if it is completely determined by some k-dimensional subspace of the input space. In this paper, we study the problem of testing whether a given n variable function $f : \mathbb{R}^n \to \{0, 1\}$, is a linear k-junta or ϵ-far from all linear k-juntas, where the closeness is measured with respect to the Gaussian measure on $\mathbb{R}^n$. Linear k-juntas are a common generalization of two fundamental classes from Boolean function analysis (both of which have been studied in property testing) **1.** k- juntas which are functions on the Boolean cube which depend on at most k of the variables and **2.** intersection of k halfspaces, a fundamental geometric concept class.

We show that the class of linear k-juntas is not testable, but adding a surface area constraint makes it testable: we give a $\mathsf{poly}(k \cdot s/\epsilon)$-query non-adaptive tester for linear k-juntas with surface area at most s. We show that the polynomial dependence on s is necessary. Moreover, we show that if the function is a linear k-junta with surface area at most s, we give a $(s \cdot k)^{O(k)}$-query non-adaptive algorithm to learn the function *up to a rotation of the basis*. In particular, this implies that we can test the class of intersections of k halfspaces in $\mathbb{R}^n$ with query complexity independent of n.

Keywords: Linear juntas; Gaussian measure; Ornstein-Uhlenbeck operator;

1. Introduction

Property testing of Boolean functions was initiated in the seminal work of Blum, Luby and Rubinfeld Blum et al. (1993) and Rubinfeld and Sudan Rubinfeld and Sudan (1996). The high level goal of property testing is the following: Given (query) access to a Boolean function f, the algorithm must distinguish between (i) the case that f belongs to a class $\mathcal{C}$ of Boolean functions (i.e., *has a property* $\mathcal{C}$), and (ii) the case that f is ϵ-far from every function belonging to $\mathcal{C}$. Here the distance between functions is measured with respect to some underlying distribution $\mathcal{D}$ and is defined as $\mathsf{dist}(f, g) = \mathbf{Pr}_{x\sim\mathcal{D}}[f(x) \neq g(x)]$. Also, the algorithm is randomized and thus only needs to succeed with high probability (as opposed to probability one). The quality of a testing algorithm

. Extended abstract. Full version appears as arxiv preprint 1806.10057v2 is measured by the number of oracle calls it makes to f – its *query complexity* – and the goal is to minimize this query complexity.

Since the works of Blum et al. (1993); Rubinfeld and Sudan (1996), property testing of Boolean functions has been a thriving field and by now several classes $\mathcal{C}$ have been studied from this perspective. These include classes such as linear functions Blum et al. (1993), low-degree polynomials Jutla et al. (2004); Bhattacharyya et al. (2010), monotonicity Fischer et al. (2002); Chakrabarty and Seshadhri (2016); Khot et al. (2015), algebraic properties Kaufman and Sudan (2008); Bhattacharyya et al. (2015, 2013) and juntas Fischer et al. (2004); Blais (2009); Chen et al. (2017b) among many others (see the surveys Ron et al. (2010); Goldreich (2017)).

Special attention has been devoted to the problem of testing juntas. Recall that a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is said to be a k-junta if f is only dependent on a subset $S \subseteq [n]$ (of size k) of the coordinates. Given (query) access to a function f, the problem of testing juntas is to decide whether f is a k-junta or ϵ-far from every k-junta (under the uniform distribution on $\{-1,1\}^n$). Some of the initial motivation Fischer et al. (2004) to study this came from the problem of long-code testing Bellare et al. (1998); Parnas et al. (2002) (related to PCPs and inapproximability). Another motivation comes from the *feature selection* problem in machine learning. It is well-known (see, e.g. Blum (1994); Blum and Langley (1997)) that learning a k-junta requires at least $\Omega(k \log n)$ samples, however k-juntas can be tested with query complexity independent of n Fischer et al. (2004).

The most obvious generalization of k-juntas to functions $f : \mathbb{R}^n \to \{-1,1\}$ is to consider functions that depend only on k of the n coordinates. However, in many statistical and machine learning models (e.g. PCA, ICA, kernel learning, dictionary learning) the choice of basis is not a priori clear. Therefore, it is natural to consider a notion of junta that is linearly invariant. We define a function $f : \mathbb{R}^n \to \{-1,1\}$ to be a *linear k-junta* if there are k unit vectors $u_1, \ldots, u_k \in \mathbb{R}^n$ and $g : \mathbb{R}^k \to \{-1,1\}$ such that $f(x) = g(\langle u_1, x\rangle, \ldots, \langle u_k, x\rangle)$.

We note that the family of linear k-juntas includes important classes of functions that have been studied in the learning and testing literature. Notably it includes:

- Boolean juntas: If $h : \{-1,1\}^n \to \{0,1\}$ is a Boolean junta, then $f(x) : \mathbb{R}^n \to \{0,1\}$ defined as $f(x) = h(\mathsf{sgn}(x_1), \ldots, \mathsf{sgn}(x_n))$ is a linear k-junta.
- Functions of halfspaces: Linear k-juntas include as a special case both halfspaces and intersections of k-halfspaces. The testability of halfspaces was studied in Matulef et al. (2009, 2010); Ron and Servedio (2015).

We consider the scenario where the ambient dimension n is large but the dimension of the relevant subspace, i.e., k is small. In this setting, we consider the following property testing question:

Question 1 *Given oracle access to a function f, is it possible to test in number of queries that depends on k (but not on n) whether f is a linear k-junta or far from all linear k-juntas?*

The problem of testing linear-juntas is closely related to the problem of *model compression* in machine learning. The goal of model compression is to take as an input a complex predictor/classifier function and to output a simpler predictor/classifier see e.g Buciluă et al. (2006). The question of model compression is extensively studied in the context of deep nets, see e.g., Ba and Caruana (2014), and follow up work, where the models are often rotationally invariant (with the caveat that the regularization often used in optimization might not be). Thus as a motivating example we may ask: given a complex deep net classifier, is there a classifier that has essentially the same performance and depends only on k of the features?

To formally state question 1 we need to define what "close" means. The standard definition is to state that f is close to g if $\mathbf{Pr}[f(x) \neq g(x)]$ is small, for some probability measure $\mathbf{Pr}$. The most natural choice of $\mathbf{Pr}$ for learning and testing functions $f : \mathbb{R}^n \to \{-1, 1\}$ is the Gaussian measure Matulef et al. (2009); Kothari et al. (2014); Neeman (2014); Balcan et al. (2012); Chen et al. (2017a); Klivans et al. (2008); Vempala (2010a); Diakonikolas et al. (2018); Balcan and Long (2013); Harsha et al. (2012). It is particularly natural in our setup since the Gaussian measure is invariant under many linear transformations, e.g., all rotations.

It is possible to show that the answer to question 1 is **no** even if $n = 2$ and $k = 1$, since without smoothness assumptions, measurable functions $f : \mathbb{R}^n \to \{-1, 1\}$ can look arbitrarily random to any finite number of queries (a more formal statement with stronger results will be discussed shortly). Since the groundbreaking work of Klivans et al. (2008), it was recognized that the surface area of a function $f : \mathbb{R}^n \to \{-1, 1\}$ is a natural complexity parameter (roughly speaking, if $A = \{x : f(x) = 1\}$, then the surface area of f is the size of the boundary of A weighted by the Gaussian measure). We therefore ask the following question:

Question 2 *Given oracle access to a function f, is it possible to test in number of queries that depends on k and s (but not on n) if f is close to any linear k-junta with surface area at most s?*

In our main result we give an affirmative answer to the question above:

Theorem 3 *There is an algorithm* Test-linear-junta *which has the following guarantee: Given oracle access to $f : \mathbb{R}^n \to \{-1, 1\}$, rank parameter k, surface area parameter s and error parameter $\epsilon > 0$, it makes $\mathsf{poly}(s, \epsilon^{-1}, k)$ queries and distinguishes between the following cases:*

1. *The function f is a linear k-junta whose surface area is at most s.*
2. *The function f is ϵ-far from any linear k-junta with surface area at most $s(1 + \epsilon)$.*

The proof can be found in the full version (and a detailed sketch is given in Section 3). We note that while the tester allows a slack of $1 + \epsilon$ in the surface area between the soundness and completeness cases, such a slack factor is required even for the easier problem of estimating surface area in $\mathbb{R}^2$ Neeman (2014). In fact, as the next theorem shows, our tester is optimal in its dependence on s up to polynomial factors (proof in the full version).

Theorem 4 *Any non-adaptive algorithm for testing whether an unknown Boolean function f is a linear 1-junta with surface area at most s versus $\Omega(1)$-far from a linear 1-junta makes at least $s^{\frac{1}{10}}$ queries.*

FINDING THE LINEAR-INVARIANT STRUCTURE

Given the previous theorem it is natural to ask for more, i.e., not just test if the function is a linear-junta but also find the junta in number of queries that depends only on k and s (but not on n). In other words, could we output $g : \mathbb{R}^k \to \{-1, 1\}$ such that there exists a projection matrix $A : \mathbb{R}^n \to \mathbb{R}^k$ and f is close to $g(Ax)$ with query complexity independent of n? We give an affirmative answer to this question:

Theorem 5 *Let $f : \mathbb{R}^n \to \{-1, 1\}$ be a linear k-junta with surface area at most s. Given the error parameter $\epsilon > 0$, the algorithm* Find-invariant-structure *makes $(s \cdot k/\epsilon)^{O(k)}$ queries and outputs $g : \mathbb{R}^k \to [-1, 1]$ so that the following holds: there exists an orthonormal set of vectors $w_1, \ldots, w_k \in \mathbb{R}^n$ such that*

$$\mathbf{E}[|f(x) - g(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle)|] = O(\epsilon).$$

Moreover, for some $g^* : \mathbb{R}^k \to \mathbb{R}$:

$$f(x) = g^*(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle).$$

Informally, the theorem states that it is possible to find the "linear-invariant" structure (i.e., the structure up to unitary transformation) of f in number of queries that depends on s and k. Of course, one cannot hope to output the relevant directions $w_1, \ldots, w_k$ explicitly as even describing these directions will require $\omega(n)$ bits of information and thus, at least those many queries. We note that the number of functions in k dimensions with $O(1)$ surface area (even up to a unitary rotation) is $\exp(\exp(k))$ and thus even our output has to be $\exp(k)$ bits. Thus, it is not possible to significantly improve on our $\exp(k \log k)$ query complexity in finding the linear-invariant structure.

TESTABILITY OF LINEAR INVARIANT FAMILIES OF LINEAR k-JUNTAS

Our ability to find the linear-invariant structure of linear k-juntas additionally allows us to test subclasses of linear k-juntas which are closed under rotation.

Definition 6 *Let $\mathcal{C}$ be any collection of functions mapping $\mathbb{R}^k$ to $\{-1, 1\}$. For any $n \in \mathbb{N}$ let:*

$\mathsf{Ind}(\mathcal{C})_n = \{f : \exists g \in \mathcal{C}$ *and orthonormal vectors $w_1, \ldots, w_k$ such that* $f(x) = g(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle).\}$

Define $\mathsf{Ind}(\mathcal{C}) = \cup_{n=k}^{\infty} \mathsf{Ind}(\mathcal{C})_n$ *and call it the* induced class of $\mathcal{C}$.

The two key properties of $\mathsf{Ind}(\mathcal{C})$ are (i) each function $f \in \mathsf{Ind}(\mathcal{C})$ is a linear k-junta, (ii) the class $\mathsf{Ind}(\mathcal{C})$ is closed under unitary transformations. The definition is a continuous analogue of the so-called "induced subclass of k-dimensional functions" from Gopalan et al. (2009) (that paper was about testing functions over $\mathsf{GF}^n[2]$). The following theorem shows that for any $\mathcal{C}$, $\mathsf{Ind}(\mathcal{C})$ is testable without any dependence on the ambient dimension.

Theorem *Let $\mathcal{C}$ be a collection of functions mapping $\mathbb{R}^k$ to $\{-1, 1\}$ such that for every $f \in \mathsf{Ind}(\mathcal{C})$,* $\mathsf{surf}(f) \le s$. *Then, there is an algorithm* Test-structure-$\mathcal{C}$ *which has the following guarantee: Given oracle access to $f : \mathbb{R}^n \to \{-1, 1\}$ and an error parameter $\epsilon > 0$, the algorithm makes $(s \cdot k/\epsilon)^{O(k)}$ queries and distinguishes between the cases (i) $f \in \mathsf{Ind}(\mathcal{C})$ and (ii) f is ϵ-far from every function $g \in \mathsf{Ind}(\mathcal{C})$.*

A particularly important instantiation of the above theorem is the following: Let $\mathcal{C}_B$ be any collection of functions mapping $\{-1, 1\}^k \to \{-1, 1\}$ and let $\mathcal{C}$ be defined as

$$\mathcal{C} = \{g : x \mapsto h(\langle w_1, x\rangle - \theta_1, \ldots, \langle w_k, x\rangle - \theta_k) |\ w_1, \ldots, w_k \in \mathbb{R}^k,\ \theta_1, \ldots, \theta_k \in \mathbb{R},\ h \in \mathcal{C}_B\}.$$

Note that $\mathcal{C}$ defined above is the set of functions obtained by composing a function from $\mathcal{C}_B$ with k-dimensional halfspaces. Consequently, $\mathsf{Ind}(\mathcal{C})$ is the of all functions which can be obtained by composing a function from $\mathcal{C}_B$ with halfspaces. As an example, if $\mathcal{C}_B$ consists of the AND function on k or fewer bits, then $\mathsf{Ind}(\mathcal{C})$ is the class of "intersections of k-halfspaces". Since the surface area of any Boolean function of k-halfspaces is bounded by $O(k)$ it follows that the this class is testable with $(k/\epsilon)^{O(k)}$ queries.

We now give a high-level description of the algorithm Test-structure-$\mathcal{C}$.

1. Run the routine Test-linear-junta with rank parameter k, surface area parameter s and error parameter $\delta > 0$ (where $\delta \approx (\epsilon/(s \cdot k))^{O(k)}$). If the test passes, go to Step 2.

2. Run the routine Find-invariant-structure with surface area parameter $2s$, rank parameter k and error parameter $\epsilon > 0$. Let $g : \mathbb{R}^\ell \to \{-1, 1\}$ be the output of this routine.

3. If the function $g : \mathbb{R}^\ell \to \{-1, 1\}$ is ϵ-close to a class $\mathcal{C}$, then accept. Else, reject.

First, observe that if the target function f passes Step 1, we are guaranteed that it is (very close to) a linear k-junta with surface area s. In Step 2, we run Find-invariant-structure. If the output of this step is g, then in Step 3, we check whether g is close to some function in $\mathsf{Ind}(\mathcal{C})_k$ and accept accordingly. Cruicially, the last step, i.e., checking whether g is close to a function in $\mathsf{Ind}(\mathcal{C})_k$ makes no queries to f.

1.1. Related Work

Testing Boolean juntas As we have already mentioned, the problem of testing juntas on $\{-1, 1\}^n$ has already been well-studied. For example, it is known Blais (2009); Chen et al. (2017b) that $\tilde{\Theta}(k^{3/2})$ queries are necessary and sufficient for non-adaptively testing k-juntas with respect to the uniform distribution, while $\tilde{\Theta}(k)$ queries are necessary and sufficient in the adaptive setting Blais et al. (2012). It even turns out to be possible to test k-juntas with respect to an unknown distribution Chen et al. (2018), although in that setting the non-adaptive query complexity becomes exponential in k. We emphasize that while the problem of junta testing inspires the problems considered in this paper, junta testing algorithms have no bearing on the problem of testing linear juntas – e.g., unlike Chen et al. (2018), there is no reason to believe that distribution-free testing of linear juntas on $\mathbb{R}^n$ is even possible, given that the space of probability measures on $\mathbb{R}^n$ is much richer than the space of probability measures on $\{-1, 1\}^n$.

Learning juntas of half-spaces. There has been extensive work on *learning* intersections and other functions of k half-spaces Blum and Kannan (1997); Vempala (2010b); Vempala and Xiao (2013); Klivans et al. (2008) . Note that these algorithms (necessarily) require time polynomial in n (whereas this work's *raison d'etre is to obtain a query complexity independent of* n). In particular, Blum and Kannan (1997) provided conditions under which intersections of halfspaces can be learnt under the uniform distribution on the ball. Vempala Vempala (2010b) extended their result to arbitrary log-concave distributions. In terms of the expressivity of the function class, Vempala and Xiao (2013) explicitly considered the problem of learning linear k-juntas (they called it subspace juntas) and showed that a linear k-junta of the form $g(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle)$ is learnable in polynomial time if the function g is identified by low moments and robust to small rotations in $\mathbb{R}^n$. Along a related but different axis, Klivans et al. (2008) showed that functions of bounded surface area in the Gaussian space are learnable in polynomial time. Finally, we remark that there also has been work in learning intersections and other functions of halfspaces over the Boolean hypercube as well Klivans et al. (2002); Gopalan et al. (2012).

Linearly Invariant Testing over Finite Fields We note that the set of linear-juntas is linearly invariant. If f is a linear k-junta and B is any $n \times n$ matrix then $x \mapsto f(Bx)$ is also a linear k-junta. Over finite fields, Kaufman and Sudan (2008) studied general criteria for when a linearly invariant property is testable, see also Bhattacharyya et al. (2013). In particular, Gopalan et al. (2009), gave a $2^{O(k)}$ query complexity algorithm to test linear juntas over finite fields. Moreover, they also show that an exponential lower bound on k is necessary. This should be contrasted with our result which shows that linear juntas over the Gaussian space can be tested with poly(k) queries.

Testing (functions) of halfspaces The question of testing halfspaces was first considered in Matulef et al. (2010) who showed that in the Gaussian space (as well as the Boolean space), halfspaces are testable with $O(1)$ queries. Subsequently, Mossel and Neeman (2015) gave a different testing algorithm for a single halfspace in the Gaussian space. In fact, Harms (2019) recently showed that halfspaces over any rotationally invariant distribution can be tested with sublinear number of queries. However, as far as we are aware, prior to our work, no non-trivial bounds were known for even testing the intersection of two halfspaces. As remarked earlier, from our work, it follows that for any arbitrary k, intersection of k-halfspaces can be tested in the Gaussian space with $\exp(k \log k)$ queries.

1.2. Techniques

The linear part $\mathcal{W}_1(f)$ of the Hermite expansion of f is approximately given by $e^{-t}(P_t f - \mathbf{E}[f])$ for large t. Here $P_t f$ is the Ornstein-Uhlenbeck operator. Both the quantities, $\mathbf{E}[f]$ and $P_t f$ can be approximated by sampling a small number of points from the Gaussian distribution and evaluating f at those points. Moreover, if $f(x) = g(\langle u_1, x\rangle, \ldots, \langle u_k, x\rangle)$ is a linear junta, then the linear part of its Hermite expansion, $\mathcal{W}_1(f)$, lies in the span of $u_1, \ldots, u_k$.

We would like to obtain "many more directions" that lie in the span of $u_1, \ldots, u_k$. We do so by considering functions of the form $f_{t,y}(x) = f(e^{-t}y + \sqrt{1-e^{-2t}}x)$, for randomly chosen y and an appropriate value of t (the experts will recognize $f_{t,y}$ as part of the definition of the Ornstein-Uhlenbeck operator). Note that $f_{t,y}$ is also a linear junta defined by the same direction $u_1, \ldots, u_k$ and therefore the linear part of the Hermite expansion of $f_{t,y}$, is also in the span of $u_1, \ldots, u_k$.

It is now natural to propose the following algorithm to test if a function is a linear k-junta: choose points y_i at random and "compute" $\mathcal{W}_1(f_{t,y_i})$ at these points. Then if the rank of the matrix spanned by $(\mathcal{W}_1(f_{t,y_i}))_i$ is at most k, then output YES; otherwise, output NO.

Of course, actually computing $\mathcal{W}_1(f_{t,y})$ requires $\mathrm{poly}(n) \gg \mathrm{poly}(k)$ samples. Instead we will approximately compute the Gram matrix

$$A_{i,j} = \langle \mathcal{W}_1(f_{t,y_i}), \mathcal{W}_1(f_{t,y_j})\rangle.$$

and test if it is close or far from a matrix of rank k. One advantage of using the Gram matrix, is that we can evaluate the entries $A_{i,j}$ by sampling random inputs to evaluate the expected values

$$\mathbf{E}[\mathcal{W}_1(f_{t,y_i})(x)\mathcal{W}_1(f_{t,y_j})(x)].$$

How do we know that $\mathcal{W}_1(f_{t,y_i})(x)$ are not very close to 0? If f has a bounded surface area then f is close to the noise stable function $P_t f$. For such noise stable functions, we prove that with good probability at a random point x, $\mathcal{W}_1(f_{t,y_i})(x)$ will be of non-negligible size. In fact, we prove much more – we show that if f is ϵ far from any linear-k-junta then for any subspace W with co-dimension at most k, it holds that for a random y with probability at least $\mathrm{poly}(\epsilon)$, the projection of $\mathcal{W}_1(f_{t,y_i})(x)$ into W will have norm at least $\mathrm{poly}(\epsilon)$. This result is later combined with a perturbation argument to establish to show that if f is ϵ-far from a linear k-junta then indeed the Gram matrix will have $k+1$ large eigenvalues. Since our analysis relies on the function f having surface area at most s, the first stage of the algorithm uses the algorithm by Neeman (2014) to test if the function of interest is of bounded surface area.

The algorithm to identify the linear invariant structure of f builds up on the ideas in the algorithm to test linear k-juntas. More precisely, we can show that if f is a linear k-junta with surface area s:

1. We can find directions $y_1, \ldots, y_\ell$ such that f is close to a function on the space spanned by the directions $\mathcal{W}_1(f_{t,y_1}), \ldots, \mathcal{W}_1(f_{t,y_\ell})$ (for some $\ell \leq k$).
2. While we cannot find $\mathcal{W}_1(f_{t,y_j})$ explicitly for any j, we can evaluate $\langle \mathcal{W}_1(f_{t,y_j}), x \rangle$ at any point x up to good accuracy.
3. With the above observation, the high level idea is to *try out all smooth functions* on the subspace spanned by $\{\langle \mathcal{W}_1(f_{t,y_1}), x\rangle, \ldots, \langle \mathcal{W}_1(f_{t,y_\ell}), x\rangle\}$. Perform *hypothesis testing* for each such function against f and output the most accurate one.

The crucial part in the above argument is that even if we have $\mathcal{W}_1(f_{t,y_1}), \ldots, \mathcal{W}_1(f_{t,y_\ell})$ implicitly, the space of "all smooth functions" on $\mathsf{span}(\langle \mathcal{W}_1(f_{t,y_1}), x\rangle, \ldots, \langle \mathcal{W}_1(f_{t,y_\ell}), x\rangle)$ has a cover whose size is independent of n. This lets us identify the linear invariant function defining f with query complexity just dependent on k and s.

In order to prove lower bounds in terms of surface area, we construct a distribution over linear 1-juntas with large surface area by splitting $\mathbb{R}^2$ into many very thin parallel strips (oriented in a random direction) and assign our function a random ± 1 value on each strip. (Note that the surface area of such a function is proportional to the number of strips.) The intuition is that no algorithm that makes non-adaptive queries can tell that such a random function is a 1-junta, because in order to "see" one of these strips, the algorithm would need to have queried multiple far-away points in a single strip. But if the number of queries is small relative to the number of strips then this is impossible – with high probability every pair of far-away query points will end up in different strips. In order to make this intuition rigorous, we also introduce a distribution on linear 2-juntas by randomly "cutting" the thin strips once in the orthogonal direction. We show that for any non-adaptive set of queries, the two distributions induce almost identical query distributions, and Yao's minimax lemma implies that no algorithm can distinguish between our random 1-juntas and our random 2-juntas.

2. Preliminaries

In this paper, unless explicitly mentioned otherwise, the domain $\mathbb{R}^n$ is always endowed with the measure γ_n, the standard n-dimensional Gaussian measure. Likewise, we will only consider functions $f \in L_2(\gamma_n)$. For such a function, and $t > 0$, we recall that the so-called Ornstein-Uhlenbeck operator P_t is defined as follows:

$$P_t f(x) = \int_y f(e^{-t}x + \sqrt{1-e^{-2t}}z)\gamma_n(z)dz$$

We will also need to recall some very basic facts about Hermite expansion for functions $f \in L_2(\gamma_n)$. In particular, recall that for all $q \geq 0$, we can define the Hermite polynomial $H_q : \mathbb{R} \to \mathbb{R}$ as

$$H_0(x) = 1;\ H_q(x) = \frac{(-1)^q}{\sqrt{q!}} \cdot e^{x^2/2} \cdot \frac{d^q}{dx^q} e^{-x^2/2}.$$

Further, for the ambient space $\mathbb{R}^n$, let us define the space $\mathcal{W}_q$ to be the linear subspace of $L_2(\gamma_n)$ spanned by $\{H_q(\langle v, x\rangle) : v \in \mathbb{S}_n\}$. Here $\mathbb{S}_n$ denotes the unit sphere in n-dimensions. For a function $g \in L_2(\gamma_n)$, we let $\widehat{g}_q : \mathbb{R}^n \to \mathbb{R}$ denote the projection of g to the subspace $\mathcal{W}_q$. Note that for any g, $\widehat{g}_q$ will be a degree-q polynomial lying in the subspace $\mathcal{W}_q$.We now recall some standard facts from Hermite analysis which can be found in any standard text on the subject (see the book by O'Donnell (2014).

Proposition 7

1. *For $q \neq q'$, the subspaces $\mathcal{W}_q$ and $\mathcal{W}_{q'}$ are orthogonal. In other words, if $r \in \mathcal{W}_q$ and $s \in \mathcal{W}_{q'}$, then $\mathbf{E}_{x\sim\gamma_n}[r(x) \cdot s(x)] = 0$.*
2. *Every function $g \in L_2(\gamma_n)$ can be expressed as $g(x) = \sum_{q\geq 0} \widehat{g}_q(x)$ where $\widehat{g}_q$ is the projection of g to $\mathcal{W}_q$.*
3. *For any $t > 0$, $(P_t g)(x) = \sum_{q\geq 0} e^{-t\cdot q} \cdot \widehat{g}_q(x)$.*

2.0.1. ORACLE COMPUTATION

We now list several useful claims which all fit the same motif: Given oracle access to $f : \mathbb{R}^n \to \mathbb{R}$, what *interesting* quantities can be computed? The proofs can all be found in the full version.

Lemma 8 *Given oracle access to $f : \mathbb{R}^n \to [-1, 1]$, error parameter $\eta > 0$, there is a function $f_{\partial,\eta} : \mathbb{R}^n \to \mathbb{R}$ such that the following holds for every $\lambda \geq 1$,*

$$\Pr_{x\sim\gamma_n}\left[\left|f_{\partial,\eta}(x) - \widehat{f}_1(x)\right| > \lambda \cdot \eta\right] \leq \lambda^{-2}.$$

Further, for any $x \in \mathbb{R}^n$, we can compute $f_{\partial,\eta}(x)$ to additive error $\pm\epsilon$ with confidence $1 - \delta$ by making $\mathsf{poly}(1/\eta, 1/\epsilon, \log(1/\delta))$ queries to the oracle for f.

Lemma 9 *Given oracle access to functions $f, g : \mathbb{R}^n \to [-1, 1]$, error parameter $\epsilon > 0$ and confidence parameter $\delta > 0$, there is an algorithm which makes $\mathsf{poly}(1/\epsilon, \log(1/\delta))$ queries to f, g and computes $\langle \widehat{f}_1, \widehat{g}_1\rangle$ up to error ϵ with confidence $1 - \delta$.*

Definition 10 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be a linear k-junta if there are at most k orthonormal vectors $u_1, \ldots, u_k \in \mathbb{R}^n$ and a function $g : \mathbb{R}^k \to \mathbb{R}$ such that*

$$f(x) = g(\langle u_1, x\rangle, \ldots, \langle u_k, x\rangle).$$

Further, if $u_1, \ldots, u_k \in W$ (a linear subspace of $\mathbb{R}^n$), then f is said to be a W-junta.

2.1. Derivatives of functions

We will use D to denote the derivative operator. In case, there are two sets of variables involved, we will explicitly indicate the variable with respect to which we are taking the derivative.

Definition 11 *For $f : \mathbb{R}^n \to \mathbb{R}$ ($f \in \mathcal{C}^\infty$) and $t \geq 0$, define the function $f_t : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,*

$$f_t(y,x) = f(e^{-t}y + \sqrt{1-e^{-2t}}x).$$

Further, in the same setting as above, we let $f_{t,y} : \mathbb{R}^n \to \mathbb{R}$,

$$f_{t,y}(x) = f(e^{-t}y + \sqrt{1-e^{-2t}}x).$$

Let D_x denote the derivative operator with respect to x and let D_y denote the derivative operator with respect to y. Then, it is easy to observe that

$$\sqrt{e^{2t}-1} \cdot D_y f_t(y,x) = D_x f_t(y,x). \tag{1}$$

Next, for a function $g : \mathbb{R}^n \to \mathbb{R}$, define $\mathcal{W}_1(g) \in \mathbb{R}^n$ as the degree-1 Hermite coefficients of g. In other words, the i^{th} coordinate of $\mathcal{W}_1(g)$

$$\mathcal{W}_1(g)[i] = \mathbf{E}[g(x) \cdot x_i],$$

where $x \sim \gamma_n$, the standard n-dimensional Gaussian measure. With respect to our earlier definition of $\widehat{g}_1$, observe that we have: $\widehat{g}_1(x) = \langle \mathcal{W}_1(g), x \rangle$. The following important lemma connects the gradient of $P_t f$ at y with $\mathcal{W}_1(f_{t,y})$.

Lemma 12

$$\mathcal{W}_1(f_{t,y}) = \sqrt{e^{2t}-1} \cdot D(P_t f)(y).$$

Lemma 13 *Given oracle access to f, noise parameter $t > 0$, error parameter $\epsilon > 0$, confidence parameter $\delta > 0$ and $y_1, y_2 \in \mathbb{R}^n$, there is an algorithm which makes $\mathsf{poly}(1/\epsilon, 1/\delta, 1/t)$ queries to f and computes $\langle D(P_t f)(y_1), D(P_t f)(y_2) \rangle$ up to error ϵ with confidence $1-\delta$.*

Proof By Lemma 12, we have

$$\langle D(P_t f)(y_1), D(P_t f)(y_2) \rangle = \frac{1}{e^{2t}-1} \cdot \langle \mathcal{W}_1(f_{t,y_1}), \mathcal{W}_1(f_{t,y_2}) \rangle.$$

We can now apply Lemma 9 to finish the proof. ■

Proposition 14 *For any $f : \mathbb{R}^n \to [-1,1]$, $\|D(P_t f)(y)\|_2 \leq (e^{2t}-1)^{-\frac{1}{2}}$.*

Proof By Lemma 12, we have $\|\mathcal{W}_1(f_{t,y})\|_2 = \sqrt{e^{2t}-1} \cdot \|D(P_t f)(y)\|_2$. Now, observe that the range of $f_{t,y}$ is $[-1,1]$ and thus, $\|\mathcal{W}_1(f_{t,y})\|_2 \leq 1$, implying the stated upper bound. ■

Lemma 15 *Given oracle access to $f : \mathbb{R}^n \to [-1,1]$, $y \in \mathbb{R}^n$, noise parameter $t > 0$, error parameter $\eta > 0$, there is a function $f_{\partial,\eta,t,y} : \mathbb{R}^n \to \mathbb{R}$ such that the following holds for every $\lambda \geq 1$,*

$$\mathbf{Pr}_{x \sim \gamma_n}[|f_{\partial,\eta,t,y}(x) - \langle D(P_t f)(y), x\rangle| > \lambda \cdot \eta] \leq \lambda^{-2}.$$

Further, for an error parameter $\epsilon > 0$, confidence parameter $\delta > 0$, we can compute $f_{\partial,t,\eta,y}$ to additive error $\pm\epsilon$ with confidence $1-\delta$ using $\mathsf{poly}(1/t, 1/\eta, 1/\epsilon, \log(1/\delta))$ queries to f.

Proof We first use Lemma 12 and obtain that

$$DP_t f(y) = \frac{1}{\sqrt{e^{2t}-1}} \cdot \mathcal{W}_1(f_{t,y}).$$

Consequently, we have that

$$\langle DP_t f(y), x\rangle = \frac{1}{\sqrt{e^{2t}-1}} \cdot \widehat{f_{t,y_1}}(x).$$

The claim now follows from Lemma 8. ■

3. Algorithm to test k-juntas

In this section, we prove Theorem 3.

Remark 16 *A convention that we shall adopt (to avoid proliferation of parameters) is to sometimes ignore the confidence parameter of the testing algorithm. Typically, whenever we can estimate a parameter within $\pm\epsilon$ with T queries with confidence $2/3$, we can do the usual "median trick" and get the same accuracy with confidence $1-\delta$ with a multiplicative $O(\log(1/\delta))$ overhead in the query complexity. Since we only need to succeed with probability 0.9 in the final algorithm, it is sufficient for each of the individual subroutines to succeed with probability sufficiently close to 1. So, unless it is crucial, at some places,we shall ignore the confidence parameter in the theorem statements and many of the calculations. It will be implicit that the confidence parameter is sufficiently close to 1.*

The algorithm Test-linear-junta is described in Figure 1. The algorithm invokes two different subroutines, Test-surface-area and Test-rank whose guarantees we state now. To do this, we first define the notion of (ϵ, s) smooth function.

Definition 17 *A function $f : \mathbb{R}^n \to \{-1,1\}$ is said to be (ϵ, s)-smooth if there is a function $g : \mathbb{R}^n \to \{-1,1\}$ such that $\mathbf{E}[|f-g|] \leq \epsilon$ and $\mathsf{surf}(g) \leq s(1+\epsilon)$.*

In other words, a function f is (ϵ, s) smooth if f is ϵ-close to some other function g (in ℓ_1 distance) and g has surface area which is essentially bounded by s. With this definition, we can now state the guarantee of the routine Test-surface-area (due to Neeman (2014)).

Theorem 18 *There is an algorithm Test-surface-area which given oracle access to a function $f : \mathbb{R}^n \to \{-1,1\}$ and error parameter $\epsilon > 0$ makes $T_{\mathsf{test}} = \mathsf{poly}(s/\epsilon)$ queries and has the following guarantee:*

1. *If f is a function with surface area at most s, then the algorithm outputs yes with probability at least $1-\epsilon$.*

Inputs

s	:=	surface area parameter
ϵ	:=	error parameter
k	:=	rank parameter

Testing algorithm

1. Run algorithm Test-surface-area with surface area parameter s and error parameter $(\epsilon/30)^4$.
2. If Test-surface-area outputs yes, then run the algorithm Test-rank with rank parameter k, surface area parameter s and error parameter ϵ.
3. If Test-rank outputs yes, then output yes. If Test-rank outputs no, output no.

Figure 1: Description of the algorithm Test-linear-junta

2. *Any function f which passes the test with probability 0.1 is (ϵ, s)-smooth.*

Next, we state the guarantee of the routine Test-rank.

Lemma 19 *The routine Test-rank has a query complexity of* $\mathsf{poly}(k, s, \epsilon^{-1})$. *Further, we have*

1. *If the function f is a linear-k-junta, then the algorithm Test-rank outputs yes with probability $1 - \epsilon$.*
2. *If $f : \mathbb{R}^n \to \{-1, 1\}$ is a $((\epsilon/30)^2, s)$-smooth function which is ϵ-far from a linear k-junta, then the algorithm Test-rank outputs no with probability $1 - \epsilon$.*

In order to prove Theorem 3, we will need the following claim which shows that property of closeness to a linear k-junta and closeness to a smooth function can be certified using a single function.

Lemma 20 *For a function $f : \mathbb{R}^n \to \{-1, 1\}$, suppose that there is a linear k-junta $g : \mathbb{R}^n \to \{-1, 1\}$ and a function $h : \mathbb{R}^n \to \{-1, 1\}$ of surface area at most s such that both g and h are ϵ-close to f. Then there is a function $\tilde{h} : \mathbb{R}^n \to \{-1, 1\}$ that is a linear k-junta* and *has surface area at most $s(1 + \sqrt{\epsilon})$, and which is $O(\sqrt{\epsilon})$-close to f.*

Proof of Theorem ??: If f is a linear k-junta with surface area at most s, then it passes both the tests Test-surface-area as well as Test-rank with probability $1 - \epsilon$. Thus, any linear k-junta with surface area at most s passes with probability at least $1-2\epsilon$ (so as long as $\epsilon \leq 0.05$, the test succeeds with probability 0.9).

On the other hand, suppose f passes Test-linear-junta with probability 0.9. Then, applying Theorem 18 is $((\epsilon/30)^4, s)$ smooth. In other words, there is a function h such that $\mathsf{surf}(h) \leq (1 + (\epsilon/30)^4) \cdot s$ which is $O(\epsilon^4)$-close to f. Further, since f passes Test-rank with probability 0.9, Lemma 19 implies that f is ϵ^2-close to some linear k-junta g. We now apply Lemma 20 to obtain that f is $O(\epsilon)$-close to some function $\tilde{h} : \mathbb{R}^n \to \{-1, 1\}$ which is a linear k-junta and $\mathsf{surf}(h) \leq (1 + O(\epsilon))s$. This concludes the proof.

■

We now turn to describing the routine Test-rank and prove Lemma 19.

Proof of Lemma 19: The bound on the query complexity of Lemma 19 is immediate from the settings of our parameters and query complexity of Lemma 13.

Input

k	:=	rank parameter
s	:=	surface area parameter
ϵ	:=	error parameter

Parameters

t	:=	$\frac{\epsilon^4}{900s^2}$
r	:=	$\frac{k \cdot s^2}{\epsilon^7}$
κ	:	$\frac{\epsilon^2}{40r}$

Testing algorithm

1. Sample directions $y_1, \dots, y_r \sim \gamma_n$.
2. Let $A_{i,j} = \langle DP_t f(y_i), DP_t f(y_j) \rangle$.
3. For all $1 \leq i, j \leq r$, compute $A_{i,j}$ up to error κ using Lemma 13. Call the estimates $B_{i,j}$.
4. For the matrix $B \in \mathbb{R}^{r \times r}$, compute the top $k+1$ singular values of B.
5. Output yes if and only if the $(k+1)^{st}$ singular value is at most $\frac{\epsilon^2}{16}$.

Figure 2: Description of the Test-rank algorithm

The first item (i.e., the completeness of Test-rank) follows from the fact that if f is a linear k-junta, $P_t f$ is also a linear k-junta. Consequently, A is a rank-k matrix. Then, A has at most k non-zero singular values. Thus, if $\sigma_1 \geq \sigma_2 \geq \dots$ are the singular values of A (in order), then $\sigma_{k+1} = 0$. By invoking Weyl's inequality, the $(k+1)^{th}$ singular value of B is at most $\epsilon^2/10$. This finishes the proof of the first item.

The proof of the second item (i.e., the soundness of Test-rank) is more involved. In particular, we can restate the second item as proving the following lemma. The proof of this lemma is highly non-trivial and appears in the full version.

Lemma 21 *Let $f : \mathbb{R}^n \to \{-1, 1\}$ be a $((\epsilon/30)^2, s)$-smooth function which is ϵ-far from a linear k-junta, then the algorithm Test-rank outputs no with probability $1 - \epsilon$.*

■

Acknowledgments

A. D. is supported by NSF CCF 1926872 (transferred from CCF 1814706). Work was primarily done when A. D. was on the faculty at Northwestern University. E. M. is partially supported by NSF award DMS-1737944 and ONR award N00014-17-1-2598. J. N. is supported by a Deutsche Forschungs-gemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy GZ 2047/1, Projekt-ID 390685813 and a Sloan Fellowship.

References

J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014.

M.-F Balcan and P. Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pages 288–316, 2013.

M.-F Balcan, E. Blais, A. Blum, and L. Yang. Active property testing. In *IEEE 53rd Annual Symposium onFoundations of Computer Science (FOCS), 2012* , pages 21–30, 2012.

Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability–towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

A. Bhattacharyya, S. Kopparty, G. Schoenebeck, M. Sudan, and D. Zuckerman. Optimal testing of reed-muller codes. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 488–497. IEEE, 2010.

A. Bhattacharyya, E. Fischer, H. Hatami, P. Hatami, and S. Lovett. Every locally characterized affine-invariant property is testable. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 429–436. ACM, 2013.

A. Bhattacharyya, E. Grigorescu, and A. Shapira. A unified framework for testing linear-invariant properties. *Random Structures & Algorithms*, 46(2):232–260, 2015.

E. Blais. Testing juntas nearly optimally. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 151–158. ACM, 2009.

Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

A. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. in AAAI Fall Symposium on ‘Relevance’, 1994.

A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces under a uniform distribution. *Journal of Computer and System Sciences*, 54(2):371–380, 1997.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comp. Sys. Sci.*, 47:549–595, 1993. Earlier version in STOC’90.

C. Buciluă, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge discovery and Data mining*, pages 535–541, 2006.

D. Chakrabarty and C. Seshadhri. An o(n) monotonicity tester for boolean functions over the hypercube. *SIAM Journal on Computing*, 45(2):461–472, 2016.

X. Chen, A. Freilich, R. Servedio, and T. Sun. Sample-based high-dimensional convexity testing. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2017a.

X. Chen, Z. Liu, Rocco A. Servedio, Y. Sheng, and J. Xie. Distribution free junta testing. In *Proceedings of the ACM STOC 2018*, 2018.

Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. In *Proceedings of the 32Nd Computational Complexity Conference*, pages 26:1–26:19, 2017b.

I. Diakonikolas, D. Kane, and A. Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1061–1073, 2018.

E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samrodnitsky. Monotonicity testing over general poset domains. In *Proc. 34th Annual ACM Symposium on the Theory of Computing*, pages 474–483, 2002.

E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *J. Computer & System Sciences*, 68(4):753–787, 2004.

Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi: 10.1017/9781108135252.

P. Gopalan, A. Klivans, and R. Meka. Learning functions of halfspaces using prefix covers. In *Conference on Learning Theory*, pages 15–1, 2012.

Parikshit Gopalan, Ryan O'Donnell, Rocco A Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. In *International Colloquium on Automata, Languages, and Programming*, pages 500–512. Springer, 2009.

N. Harms. Testing halfspaces over rotationally invariant distributions. In *Proceedings of SODA 2019*, 2019.

P. Harsha, A. Klivans, and R. Meka. An invariance principle for polytopes. *Journal of the ACM (JACM)*, 59(6):29, 2012.

Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. In *Proc. 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 423–432. IEEE Computer Society Press, 2004.

T. Kaufman and M. Sudan. Algebraic property testing: the role of invariance. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 403–412, 2008.

S. Khot, D. Minzer, and M. Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 52–58. IEEE, 2015.

A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 177–186, 2002.

A. Klivans, R. O'Donnell, and R. Servedio. Learning geometric concepts via Gaussian surface area. In *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 541–550, 2008.

P. Kothari, A. Nayyeri, R. O'Donnell, and C. Wu. Testing surface area. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1204–1214. SIAM, 2014.

K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. In *Proc. 20th Annual Symposium on Discrete Algorithms (SODA)*, 2009.

K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. *SIAM J. on Comput.*, 39(5):2004–2047, 2010.

E. Mossel and J. Neeman. Robust optimality of gaussian noise stability. *Journal of the European Mathematical Society*, 17(2):433–482, 2015.

J. Neeman. Testing surface area with arbitrary accuracy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 393–397. ACM, 2014.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

M. Parnas, D. Ron, and A. Samorodnitsky. Testing basic boolean formulae. *SIAM J. Disc. Math.*, 16:20–46, 2002. URL `citeseer.ifi.unizh.ch/parnas02testing.html`.

D. Ron and R. Servedio. Exponentially Improved Algorithms and Lower Bounds for Testing Signed Majorities. *Algorithmica*, 72(2):400–429, 2015.

Dana Ron et al. Algorithmic and analysis techniques in property testing. *Foundations and Trends® in Theoretical Computer Science*, 5(2):73–205, 2010.

R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. on Comput.*, 25:252–271, 1996.

S. Vempala. Learning convex concepts from gaussian distributions with PCA. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 124–130, 2010a.

S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *Journal of the ACM (JACM)*, 57(6):32, 2010b.

S. Vempala and Y. Xiao. Complexity of learning subspace juntas and ICA. In *2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, November 3-6, 2013*, pages 320–324, 2013.