


Downsampling for Testing and Learning in Product Distributions

Nathaniel Harms  

University of Waterloo, Canada

Yuichi Yoshida  

National Institute of Informatics, Tokyo, Japan

Abstract

We study distribution-free property testing and learning problems where the unknown probability distribution is a product distribution over \mathbb{R}^d . For many important classes of functions, such as intersections of halfspaces, polynomial threshold functions, convex sets, and k -alternating functions, the known algorithms either have complexity that depends on the support size of the distribution, or are proven to work only for specific examples of product distributions. We introduce a general method, which we call *downsampling*, that resolves these issues. Downsampling uses a notion of “rectilinear isoperimetry” for product distributions, which further strengthens the connection between isoperimetry, testing and learning. Using this technique, we attain new efficient distribution-free algorithms under product distributions on \mathbb{R}^d :

1. A simpler proof for non-adaptive, one-sided monotonicity testing of functions $[n]^d \rightarrow \{0, 1\}$, and improved sample complexity for testing monotonicity over unknown product distributions, from $O(d^7)$ [Black, Chakrabarty, & Seshadhri, SODA 2020] to $\tilde{O}(d^3)$.
2. Polynomial-time agnostic learning algorithms for functions of a constant number of halfspaces, and constant-degree polynomial threshold functions;
3. An $\exp(O(d \log(dk)))$ -time agnostic learning algorithm, and an $\exp(O(d \log(dk)))$ -sample tolerant tester, for functions of k convex sets; and a $2^{\tilde{O}(d)}$ sample-based one-sided tester for convex sets;
4. An $\exp(\tilde{O}(k\sqrt{d}))$ -time agnostic learning algorithm for k -alternating functions, and a sample-based tolerant tester with the same complexity.

2012 ACM Subject Classification Mathematics of computing \rightarrow Probabilistic algorithms; Theory of computation \rightarrow Machine learning theory; Theory of computation \rightarrow Streaming, sublinear and near linear time algorithms

Keywords and phrases property testing, learning, monotonicity, halfspaces, intersections of halfspaces, polynomial threshold functions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.71

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: <https://arxiv.org/abs/2007.07449>

Funding *Nathaniel Harms*: Research funded partly by NSERC. Some of this work was done while the author was visiting NII, Tokyo.

Yuichi Yoshida: Supported in part by JSPS KAKENHI Grant Number 18H05291 and 20H05965.

1 Introduction

In property testing and learning, the goal is to design algorithms that use as little information as possible about the input while still being correct (with high probability). This includes using as little information as possible about the probability distribution against which correctness is measured. Information about the probability distribution could be in the form



© Nathaniel Harms and Yuichi Yoshida;

licensed under Creative Commons License CC-BY 4.0

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).

Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;

Article No. 71; pp. 71:1–71:19



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



of guarantees on this distribution (e.g. it is guaranteed to be uniform, or Gaussian), or in the form of samples from the distribution. So we want to minimize the requirements on this distribution, as well as the number of samples used by the algorithm.

Progress on high-dimensional property testing and learning problems is usually made by studying algorithms for the uniform distribution over the hypercube $\{\pm 1\}^d$, or the standard Gaussian distribution over \mathbb{R}^d , as the simplest case. For example, efficiently learning intersections of halfspaces is a major open problem in learning theory [23, 33], and progress on this problem has been made by studying the uniform distribution over the hypercube $\{\pm 1\}^d$ and the Gaussian as special cases [30, 34, 38]. Another important example is the class of degree- k polynomial threshold functions (PTFs). Unlike intersections of halfspaces, these can be efficiently learned in the PAC model [33], but *agnostic* learning is more challenging. Again, progress has been made by studying the hypercube [22]. An even more extreme example is the class of convex sets, which are not learnable in the distribution-free PAC model, because they have infinite VC dimension, but which become learnable under the Gaussian [34]. The uniform distribution over the hypercube and the Gaussian are both examples of *product distributions*, so the next natural question to ask is, can these results be generalized to any *unknown* product distribution? A partial answer was given by Blais, O’Donnell, & Wimmer [10] for some of these classes; in this paper we resolve this question.

Similar examples appear in the property testing literature. Distribution-free property testing and testing functions with domain \mathbb{R}^d are emerging trends in the field (e.g. [2, 21, 29, 19, 26, 9]). Testing monotonicity is one of the most well-studied problems in property testing, and recent work [6] has extended this study to product distributions over domain \mathbb{R}^d . Work of Chakrabarty & Seshadhri [17], Khot, Minzer, & Safra [32], and Black, Chakrabarty, & Seshadhri [5, 6] has resulted in efficient $o(d)$ -query algorithms for the hypercube $\{\pm 1\}^d$ [32] and the hypergrid $[n]^d$. Black, Chakrabarty, & Seshadhri [6] showed that testing monotonicity over unknown product distributions on \mathbb{R}^d could be done with $\tilde{O}(d^{5/6})$ queries and $O(d^7)$ samples. Their “domain reduction” method is intricate and specialized for the problem of testing monotonicity. We improve¹ the sample complexity to $\tilde{O}(d^3)$ using a much simpler proof. We also generalize the testers of [18, 15] for convex sets and k -alternating functions, respectively, and provide new testers for arbitrary functions of convex sets.

This paper provides a general framework for designing distribution-free testing and learning algorithms under product distributions on \mathbb{R}^d , which may be finite or continuous. An algorithm is *distribution-free under product distributions* if it does not require any prior knowledge of the probability distribution, except the guarantee that it is a product distribution. The technique in this paper, which we call *downsampling*, improves upon previous methods (in particular, [6, 10]), in a few ways. It is more general and does not apply only to a specific type of algorithm [10] or a specific problem [6], and we use it to obtain many other results. It is conceptually simpler. And it allows quantitative improvements over both [10] and [6].

Organization

This paper is presented as an extended abstract, with the results, techniques, and definitions described in the main text, and most of the proofs given in the full version of the paper. We present our result for testing monotonicity in this extended abstract, as an example application of our techniques. In Section 1.1, we describe the main results of this paper in

¹ An early version of this paper proved a weaker result, with two-sided error and worse sample complexity.

context of the related work. In Section 1.2, we briefly describe the main techniques in the paper. Section 2 presents the definitions and lemmas required by the main results. Section 3 gives the proofs for our results on testing monotonicity. The remaining proofs are in the full version. For simplicity, continuous distributions are treated in the main text and the method for extending the results to finite distributions are handled separately.

1.1 Results

See Table 1 for a summary of our results on property testing, and Table 2 for a summary of our results on learning. Some standard definitions are as follows.

For a set \mathcal{P} of distributions over X and a set \mathcal{H} of functions $X \rightarrow \{\pm 1\}$, a *distribution-free* property testing algorithm for \mathcal{H} under \mathcal{P} is a randomized algorithm that is given a parameter $\epsilon > 0$. It has access to the input probability distribution $\mathcal{D} \in \mathcal{P}$ via a *sample oracle*, which returns an independent sample from \mathcal{D} . It has access to the input function $f : X \rightarrow \{\pm 1\}$ via a *query oracle*, which given query $x \in X$ returns the value $f(x)$. A *two-sided* distribution-free testing algorithm must satisfy:

1. If $f \in \mathcal{H}$ then the algorithm accepts with probability at least $2/3$;
2. If f is ϵ -far from \mathcal{H} with respect to μ then the algorithm rejects with probability at least $2/3$.

A *one-sided* algorithm must accept with probability 1 when $f \in \mathcal{H}$. An (ϵ_1, ϵ_2) -tolerant tester must accept with probability at least $2/3$ when $\exists h \in \mathcal{H}$ such that $\mathbb{P}_{x \sim \mu} [f(x) \neq h(x)] \leq \epsilon_1$ and reject when f is ϵ_2 -far from \mathcal{H} with respect to μ .

In the *query model*, the queries to the query oracle can be arbitrary. In the *sample model*, the tester queries a point $x \in X$ if and only if x was obtained from the sample oracle. A tester in the query model is *adaptive* if it makes its choice of query based on the answers to previous queries. It is *non-adaptive* if it chooses its full set of queries in advance, before obtaining any of the answers. The *sample complexity* of an algorithm is the number of samples requested from the sample oracle. The *query complexity* of an algorithm is the number of queries made to the query oracle.

Let \mathcal{H} be a set of functions $X \rightarrow \{\pm 1\}$ and let \mathcal{P} be a set of probability distributions over X . A learning algorithm for \mathcal{H} under \mathcal{P} (in the *non-agnostic* or *realizable*) model is a randomized algorithm that receives a parameter $\epsilon > 0$ and has *sample access* to an input function $f \in \mathcal{H}$. Sample access means that the algorithm may request an independent random example $(x, f(x))$ where x is sampled from some input distribution $\mathcal{D} \in \mathcal{P}$. The algorithm is required to output a function $g : X \rightarrow \{\pm 1\}$ that, with probability $2/3$, satisfies the condition $\mathbb{P}_{x \sim \mathcal{D}} [f(x) \neq g(x)] \leq \epsilon$.

In the *agnostic* setting, the algorithm instead has sample access to an input distribution \mathcal{D} over $X \times \{0, 1\}$ whose marginal over X is in \mathcal{P} (i.e. it receives samples of the form $(x, b) \in X \times \{0, 1\}$). The algorithm is required to output a function $g : X \rightarrow \{\pm 1\}$ that, with probability $2/3$, satisfies the following condition: $\forall h \in \mathcal{H}$,

$$\mathbb{P}_{(x,b) \sim \mathcal{D}} [g(x) \neq b] \leq \mathbb{P}_{(x,b) \sim \mathcal{D}} [h(x) \neq b] + \epsilon.$$

A *proper* learning algorithm is one whose output must also satisfy $g \in \mathcal{H}$; otherwise it is *improper*.

1.1.1 Testing Monotonicity

Testing monotonicity is the problem of testing whether an unknown function $f : X \rightarrow \{0, 1\}$ is monotone, where X is a partial order. It is one of the most commonly studied problems in the field of property testing. Previous work on this problem has mostly focused on uniform

probability distributions (exceptions include [1, 28, 16, 9]) and finite domains. However, there is growing interest in property testing for functions on domain \mathbb{R}^d ([2, 21, 29, 19, 26, 9]) and [6] generalized the problem to this domain.

Testing monotonicity under product distributions has been studied a few times. Ailon & Chazelle [1] gave a distribution-free monotonicity tester for real-valued functions under product distributions on $[n]^d$, with query complexity $O(\frac{1}{\epsilon} d 2^d \log n)$. Chakrabarty *et al.* [16] improved this to $O(\frac{1}{\epsilon} d \log n)$ and gave a matching lower bound. This lower bound applies to the *real-valued* case. For the *boolean-valued* case, monotonicity testers under the uniform distribution on $\{\pm 1\}^d$ [17, 32] and $[n]^d$ [5, 6] are known with query complexity $o(d)$. In [6], an $o(d)$ -query tester was given for domain \mathbb{R}^d . That paper showed that there is a one-sided, non-adaptive, distribution-free monotonicity tester under product distributions on \mathbb{R}^d , with query complexity $O\left(\frac{d^{5/6}}{\epsilon^{4/3}} \text{poly log}(d/\epsilon)\right)$ and sample complexity $O((d/\epsilon)^7)$. In this paper we improve the sample complexity to $\tilde{O}((d/\epsilon)^3)$, while greatly simplifying the proof.

► **Theorem 1.1.** *There is a one-sided, non-adaptive ϵ -tester for monotonicity of functions $\mathbb{R}^d \rightarrow \{0, 1\}$ that is distribution-free under (finite or continuous) product distributions, using*

$$O\left(\frac{d^{5/6}}{\epsilon^{4/3}} \text{poly log}(d/\epsilon)\right)$$

queries and $O(\frac{d^3}{\epsilon^3} \log(d/\epsilon))$ samples.

The main result of [6] is a “domain reduction” lemma, which shows that for any function $f : [n]^d \rightarrow \{0, 1\}$, the distance to monotonicity (under the uniform distribution) is not significantly reduced by sampling a random subgrid S of $[n]^d$ with sides of length $k = O(d^7)$ and restricting f to the domain S . To prove this lemma, [6] develops specialized structural tools for analyzing the “violation graph” of f . The violation graph is a standard object in the study of testing monotonicity. Its vertices are points in the domain, and its edges are “violations of monotonicity”: pairs of points $x \prec y$ in the partial order where $f(x) > f(y)$. The distance of f to monotonicity is related to the size of the maximum matching in this graph (due to a result of [25]). The main technical challenge of [6] is to show how to find large matchings in the violation graph under the random restriction to a subgrid, for which they do a “line-by-line analysis” to show how to preserve many of the matched endpoints on each line in the grid. Compared to the technique in our paper, their proof is highly specialized to testing monotone functions, and requires a much more technical analysis. Our result replaces this domain reduction method with a simpler and more general 2-page argument, and gives a different generalization to the distribution-free case. See Section 3 for the proofs.

1.1.2 Learning Functions of Halfspaces

Intersections of k halfspaces have VC dimension $\Theta(dk \log k)$ [14, 20], so the sample complexity of learning is known, but it is not possible to efficiently find k halfspaces whose intersection is correct on the sample, unless $P = NP$ [13]. Therefore the goal is to find efficient “improper” algorithms that output a function other than an intersection of k halfspaces. Several learning algorithms for intersections of k halfspaces actually work for arbitrary functions of k halfspaces. We will write \mathcal{B}_k for the set of all functions $\{0, 1\}^k \rightarrow \{0, 1\}$, and for any class \mathcal{F} of functions we will write $\mathcal{B}_k \circ \mathcal{F}$ as the set of all functions $x \mapsto g(f_1(x), \dots, f_k(x))$ where $g \in \mathcal{B}_k$ and each $f_i \in \mathcal{F}$. Then for \mathcal{H} the class of halfspaces, Klivans, O’Donnell, & Servedio [33] gave a (non-agnostic) learning algorithm for $\mathcal{B}_k \circ \mathcal{H}$ over the uniform distribution on $\{\pm 1\}^d$ with complexity $d^{O(k^2/\epsilon^2)}$, Kalai, Klivans, Mansour, & Servedio [30] presented an agnostic algorithm with complexity $d^{O(k^2/\epsilon^4)}$ in the same setting using “polynomial regression”.

■ **Table 1** Testing results.

	$\text{unif}(\{\pm 1\}^d)$	$\text{unif}([n]^d)$	Gaussian	\forall Products
1-Sided Testing Monotonicity (Query model)	$\tilde{O}\left(\frac{\sqrt{d}}{\epsilon^2}\right)$ [32]	$\tilde{O}\left(\frac{d^{5/6}}{\epsilon^{4/3}}\right)$ [6]	$\tilde{O}\left(\frac{d^{5/6}}{\epsilon^{4/3}}\right)$ [6]	$\tilde{O}\left(\frac{d^{5/6}}{\epsilon^{4/3}}\right)$ queries, $\tilde{O}\left(\left(\frac{d}{\epsilon}\right)^3\right)$ samples (Thm. 1.1)
1-Sided Testing Convex Sets (Sample model)	–	–	$\left(\frac{d}{\epsilon}\right)^{(1+o(1))d}$ $2^{\Omega(d)}$ [18]	$\left(\frac{d}{\epsilon}\right)^{(1+o(1))d}$ (Thm. 1.4)
Tolerant Testing Functions of k Convex Sets (Sample model)	–	–	–	$\left(\frac{dk}{\epsilon}\right)^{O(d)}$ (Thm. 1.5)
Tolerant Testing k -Alternating Functions (Sample model)	–	$\left(\frac{dk}{\tau}\right)^{O\left(\frac{k\sqrt{d}}{\tau^2}\right)}$ $\tau = \epsilon_2 - 3\epsilon_1$ [15]	–	$\left(\frac{dk}{\tau}\right)^{O\left(\frac{k\sqrt{d}}{\tau^2}\right)}$ $\tau = \epsilon_2 - \epsilon_1$ (Thm. 1.8)

Polynomial regression is a powerful technique, so it is important to understand when it can be applied. Blais, O’Donnell, & Wimmer [10] studied how to generalize it to arbitrary product distributions. With their method, they obtained an agnostic learning algorithm for $\mathcal{B}_k \circ \mathcal{H}$ with complexity $(dn)^{O(k^2/\epsilon^4)}$ for product distributions $X_1 \times \dots \times X_d$ where each $|X_i| = n$, and complexity $d^{O(k^2/\epsilon^4)}$ for the “polynomially bounded” continuous distributions. This is not a complete generalization, because, for example, on the grid $[n]^d$ its complexity depends on n . This prevents a full generalization to the domain \mathbb{R}^d . Their algorithm also requires some prior knowledge of the support or support size. We use a different technique and fully generalize the polynomial regression algorithm to arbitrary product distributions. See the full version for the proof.

► **Theorem 1.2.** *There is an improper agnostic learning algorithm for $\mathcal{B}_k \circ \mathcal{H}$, which is distribution-free under (continuous or finite) product distributions over \mathbb{R}^d , with time complexity*

$$\min \left\{ \left(\frac{dk}{\epsilon}\right)^{O\left(\frac{k^2}{\epsilon^4}\right)}, O\left(\frac{1}{\epsilon^2} \left(\frac{3dk}{\epsilon}\right)^d\right) \right\}.$$

1.1.3 Learning Polynomial Threshold Functions

Degree- k PTFs are another generalization of halfspaces. A function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ is a degree- k PTF if there is a degree- k polynomial $p : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $f(x) = \text{sign}(p(x))$. Degree- k PTFs can be PAC learned in time $d^{O(k)}$ using linear programming [33], but agnostic learning is more challenging. Diakonikolas *et al.* [22] previously gave an agnostic learning algorithm for degree- k PTFs in the uniform distribution over $\{\pm 1\}^d$ with time complexity $d^{\psi(k,\epsilon)}$, where

$$\psi(k, \epsilon) := \min \left\{ O(\epsilon^{-2^{k+1}}), 2^{O(k^2)} (\log(1/\epsilon)/\epsilon^2)^{4k+2} \right\}.$$

The main result of that paper is an upper bound on the noise sensitivity of PTFs. Combined with the reduction of [10], this implies an algorithm for the uniform distribution over $[n]^d$ with complexity $(dn)^{\psi(k,\epsilon)}$ and for the Gaussian distribution with complexity $d^{\psi(k,\epsilon)}$.

■ **Table 2** *Learning results.* All learning algorithms are agnostic except that of [38]. The PTF result for the Gaussian follows from the two cited works but is not stated in either. All statements are informal, see references for restrictions and qualifications. For PTFs, $\psi(k, \epsilon) := \min \left\{ O(\epsilon^{-2k+1}), 2^{O(k^2)} (\log(1/\epsilon)/\epsilon^2)^{4k+2} \right\}$.

	$\text{unif}(\{\pm 1\}^d)$	$\text{unif}([n]^d)$	Gaussian	\forall Products
Functions of k Convex Sets	$\Omega(2^d)$	–	$d^{O\left(\frac{\sqrt{d}}{\epsilon^4}\right)}, 2^{\Omega(\sqrt{d})}$ [34]	$O\left(\frac{1}{\epsilon^2} \left(\frac{6dk}{\epsilon}\right)^d\right)$ (Thm. 1.6)
Functions of k Halfspaces	$d^{O\left(\frac{k^2}{\epsilon^4}\right)}$ [30]	$(dn)^{O\left(\frac{k^2}{\epsilon^4}\right)}$ [10]	$d^{O\left(\frac{\log k}{\epsilon^4}\right)},$ $\text{poly}\left(d, \left(\frac{k}{\epsilon}\right)^k\right)$ [34, 38] (Intersections only)	$\left(\frac{dk}{\epsilon}\right)^{O\left(\frac{k^2}{\epsilon^4}\right)}$ (Thm. 1.2)
Degree- k PTFs	$d^{\psi(k, \epsilon)}$ [22]	$(dn)^{\psi(k, \epsilon)}$ [22, 10]	$d^{\psi(k, \epsilon)}$ [22, 10]	$\left(\frac{dk}{\epsilon}\right)^{\psi(k, \epsilon)}$ (Thm. 1.3)
k -Alternating Functions	$2^{\Theta\left(\frac{k\sqrt{d}}{\epsilon}\right)}$ [8]	$\left(\frac{dk}{\tau}\right)^{O\left(\frac{k\sqrt{d}}{\tau^2}\right)}$ (Testing) [15]	–	$\left(\frac{dk}{\epsilon}\right)^{O\left(\frac{k\sqrt{d}}{\epsilon^2}\right)}$ (Thm. 1.7)

Our agnostic learning algorithm for degree- k PTFs eliminates the dependence on n and works for any unknown product distribution over \mathbb{R}^n , while matching the complexity of [22] for the uniform distribution over the hypercube. See the full version for the proof.

► **Theorem 1.3.** *There is an improper agnostic learning algorithm for degree- k PTFs, which is distribution-free under (finite or continuous) product distributions over \mathbb{R}^d , with time complexity*

$$\min \left\{ \left(\frac{kd}{\epsilon}\right)^{\psi(k, \epsilon)}, O\left(\frac{1}{\epsilon^2} \left(\frac{9dk}{\epsilon}\right)^d\right) \right\}.$$

1.1.4 Testing & Learning Convex Sets

One of the first properties (sets) of functions $\mathbb{R}^d \rightarrow \{0, 1\}$ to be studied in the property testing literature is the set of indicator functions of convex sets, i.e. functions $f : \mathbb{R}^d \rightarrow \{0, 1\}$ where $f^{-1}(1)$ is convex. Write \mathcal{C} for this class of functions. This problem has been studied in various models of testing [36, 35, 18, 4, 7]. In this paper we consider the *sample-based* model of testing, where the tester receives only random examples of the function and cannot make queries. This model of testing has received a lot of recent attention (e.g. [2, 4, 12, 18, 27, 29, 37, 9]), partly because it matches the standard sample-based model for learning algorithms.

Chen *et al.* [18] gave a sample-based tester for \mathcal{C} under the Gaussian distribution on \mathbb{R}^d with one-sided error and sample complexity $(d/\epsilon)^{O(d)}$, along with a lower bound (for one-sided testers) of $2^{\Omega(d)}$. We match their upper bound while generalizing the tester to be distribution-free under product distributions. See the full version for proofs.

► **Theorem 1.4.** *There is a sample-based one-sided ϵ -tester for \mathcal{C} which is distribution-free under (finite or continuous) product distributions that uses at most $O\left(\left(\frac{6d}{\epsilon}\right)^d\right)$ samples.*

A more powerful kind of tester is an (ϵ_1, ϵ_2) -tolerant tester, which must accept (with high probability) any function that is ϵ_1 -close to the property, while rejecting functions that are ϵ_2 -far. Tolerantly testing convex sets has been studied by [3] for the uniform distribution

over the 2-dimensional grid, but not (to the best of our knowledge) in higher dimensions. We obtain a sample-based tolerant tester (and distance) approximator for convex sets in high dimension. In fact, recall that \mathcal{B}_k is the set of all functions $\{0, 1\}^k \rightarrow \{0, 1\}$ and $\mathcal{B}' \subset \mathcal{B}_k$ any subset, so $\mathcal{B}' \circ \mathcal{C}$ is any property of functions of convex sets. We obtain a distance approximator for any such property:

► **Theorem 1.5.** *Let $\mathcal{B}' \subset \mathcal{B}_k$. There is a sample-based algorithm, which is distribution-free under (finite or continuous) product distributions, that approximates distance to $\mathcal{B}' \circ \mathcal{C}$ up to additive error ϵ using $O\left(\frac{1}{\epsilon^2} \left(\frac{3dk}{\epsilon}\right)^d\right)$ samples. Setting $\epsilon = (\epsilon_2 - \epsilon_1)/2$ we obtain an (ϵ_1, ϵ_2) -tolerant tester with sample complexity $O\left(\frac{1}{(\epsilon_2 - \epsilon_1)^2} \left(\frac{6dk}{\epsilon_2 - \epsilon_1}\right)^d\right)$.*

General distribution-free learning of convex sets is not possible, since this class has infinite VC dimension. However, they can be learned under the Gaussian distribution. Non-agnostic learning under the Gaussian was studied by Vempala [38, 39]. Agnostic learning under the Gaussian was studied by Klivans, O'Donnell, & Servedio [34] who presented a learning algorithm with complexity $d^{O(\sqrt{d}/\epsilon^4)}$, and a lower bound of $2^{\Omega(\sqrt{d})}$.

Unlike the Gaussian, there is a trivial lower bound of $\Omega(2^d)$ in arbitrary product distributions, because any function $f : \{\pm 1\}^d \rightarrow \{0, 1\}$ belongs to this class. However, unlike the general distribution-free case, we show that convex sets (or any functions of convex sets) can be learned under unknown product distributions.

► **Theorem 1.6.** *There is an agnostic learning algorithm for $\mathcal{B}_k \circ \mathcal{C}$, which is distribution-free under (finite or continuous) product distributions over \mathbb{R}^d , with time complexity $O\left(\frac{1}{\epsilon^2} \cdot \left(\frac{6dk}{\epsilon}\right)^d\right)$.*

1.1.5 Testing & Learning k -Alternating Functions

A k -alternating function $f : X \rightarrow \{\pm 1\}$ on a partial order X is one where for any chain $x_1 < \dots < x_m$ in X , f changes value at most k times. Learning k -alternating functions on domain $\{\pm 1\}^d$ was studied by Blais *et al.* [8], motivated by the fact that these functions are computed by circuits with few negation gates. They show that $2^{\Theta(k\sqrt{d}/\epsilon)}$ samples are necessary and sufficient in this setting. Canonne *et al.* [15] later obtained an algorithm for (ϵ_1, ϵ_2) -tolerant testing k -alternating functions, when $\epsilon_2 > 3\epsilon_1$, in the uniform distribution over $[n]^d$, with query complexity $(kd/\tau)^{O(k\sqrt{d}/\tau^2)}$, where $\tau = \epsilon_2 - 3\epsilon_1$.

We obtain an agnostic learning algorithm for k -alternating functions that matches the query complexity of the tester in [15], and nearly matches the complexity of the (non-agnostic) learning algorithm of [8] for the uniform distribution over the hypercube. See the full version for proofs.

► **Theorem 1.7.** *There is an agnostic learning algorithm for k -alternating functions, which is distribution-free under (finite or continuous) product distributions over \mathbb{R}^d , that runs in time at most*

$$\min \left\{ \left(\frac{dk}{\epsilon}\right)^{O\left(\frac{k\sqrt{d}}{\epsilon^2}\right)}, O\left(\frac{1}{\epsilon^2} \left(\frac{3kd}{\epsilon}\right)^d\right) \right\}.$$

We also generalize the tolerant tester of [15] to be distribution-free under product distributions, and eliminate the condition $\epsilon_2 > 3\epsilon_1$.

► **Theorem 1.8.** *For any $\epsilon_2 > \epsilon_1 > 0$, let $\tau = (\epsilon_2 - \epsilon_1)/2$, there is a sample-based (ϵ_1, ϵ_2) -tolerant tester for k -alternating functions using $\left(\frac{dk}{\tau}\right)^{O\left(\frac{k\sqrt{d}}{\tau^2}\right)}$ samples, which is distribution-free under (finite or continuous) product distributions over \mathbb{R}^d .*

1.2 Techniques

What connects these diverse problems is a notion of rectilinear surface area or isoperimetry that we call “block boundary size”. There is a close connection between learning & testing and various notions of isoperimetry or surface area (e.g. [17, 33, 34, 32]). We show that testing or learning a class \mathcal{H} on product distributions over \mathbb{R}^d can be reduced to testing and learning on the *uniform* distribution over $[r]^d$, where r is determined by the block boundary size, and we call this reduction “downsampling”. The name *downsampling* is used in image and signal processing for the process of reducing the resolution of an image or reducing the number of discrete samples used to represent an analog signal. We adopt the name because our method can be described by analogy to image or signal processing as the following 2-step process:

1. Construct a “digitized” or “pixellated” image of the function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ by sampling from the distribution and constructing a grid in which each cell has roughly equal probability mass; and
2. Learn or test the “low-resolution” pixellated function.

As long as the function f takes a constant value in the vast majority of “pixels”, the low resolution version seen by the algorithm is a good enough approximation for testing or learning. The block boundary size is, informally, the number of pixels on which f is not constant.

This technique reduces distribution-free testing and learning problems to the uniform distribution in a way that is conceptually simpler than in the prior work [10, 6]. However, some technical challenges remain. The first is that it is not always easy to bound the number of “pixels” on which a function f is not constant – for example, for PTFs. Second, unlike in the uniform distribution, the resulting downsampled function class on $[r]^d$ is not necessarily “the same” as the original class – for example, halfspaces on \mathbb{R}^d are not downsampled to halfspaces on $[r]^d$, since the “pixels” are not of equal size. Thus, geometric arguments may not work, unlike the case for actual images.

A similar technique of constructing “low-resolution” representations of the input has been used and rediscovered ad-hoc a few times in the property testing literature, but only for the uniform distribution over $[n]^d$ [31, 36, 24, 12, 15], or the Gaussian in [18]. With this paper, we aim to provide a unified and generalized study of this simple and powerful technique.

1.3 Block Boundary Size

Informally, we define the *r-block boundary size* $\text{bbs}(\mathcal{H}, r)$ of a class \mathcal{H} of functions $\mathbb{R}^d \rightarrow \{0, 1\}$ as the maximum number of grid cells on which a function $f \in \mathcal{H}$ is non-constant, over all possible $r \times \dots \times r$ grid partitions of \mathbb{R}^d (which are not necessarily evenly spaced) – see Section 2 for formal definitions. Whether downsampling can be applied to \mathcal{H} depends on whether

$$\lim_{r \rightarrow \infty} \frac{\text{bbs}(\mathcal{H}, r)}{r^d} \rightarrow 0,$$

and the complexity of the algorithms depends on how large r must be for the non-constant blocks to vanish relative to the whole r^d grid. A general observation is that any function class \mathcal{H} where downsampling can be applied can be learned under unknown product distributions with a finite number of samples; for example, this holds for convex sets even though the VC dimension is infinite.

► **Proposition 1.9.** *Let \mathcal{H} be any set of functions $\mathbb{R}^d \rightarrow \{0, 1\}$ (measurable with respect to continuous product distributions) such that*

$$\lim_{r \rightarrow \infty} \frac{\text{bbs}(\mathcal{H}, r)}{r^d} = 0.$$

Then there is some function $\sigma(d, \epsilon)$ such that \mathcal{H} is distribution-free learnable under product distributions, up to error ϵ , with $\sigma(d, \epsilon)$ samples.

For convex sets, monotone functions, k -alternating functions, and halfspaces, $\text{bbs}(\mathcal{H}, r)$ is easy to calculate. For degree- k PTFs, it is more challenging – it requires proving a bound on the number of unevenly-spaced grid cells in \mathbb{R}^d in which a degree- k multivariate polynomial might take the value 0; this result may be of independent interest.

We obtain this result by proving a more general lemma. We say that a function $f : \mathbb{R}^d \rightarrow \{0, 1\}$ induces a connected component S if for every $x, y \in S$ there is a continuous curve in \mathbb{R}^d from x to y such that $f(z) = f(x) = f(y)$ for all z on the curve, and S is a maximal such set. Then we prove a general lemma that bounds the block boundary size by the number of connected components induced by functions $f \in \mathcal{H}$.

► **Lemma 1.10** (Informal, see full version). *Suppose that for any axis-aligned affine subspace A of affine dimension $n \leq d$, and any function $f \in \mathcal{H}$, f induces at most k^n connected components in A . Then for $r = \Omega(dk/\epsilon)$, $\text{bbs}(\mathcal{H}, r) \leq \epsilon \cdot r^d$.*

This lemma in fact generalizes all computations of block boundary size in this paper (up to constant factors in r). Using a theorem of Warren [40], we get the following corollary:

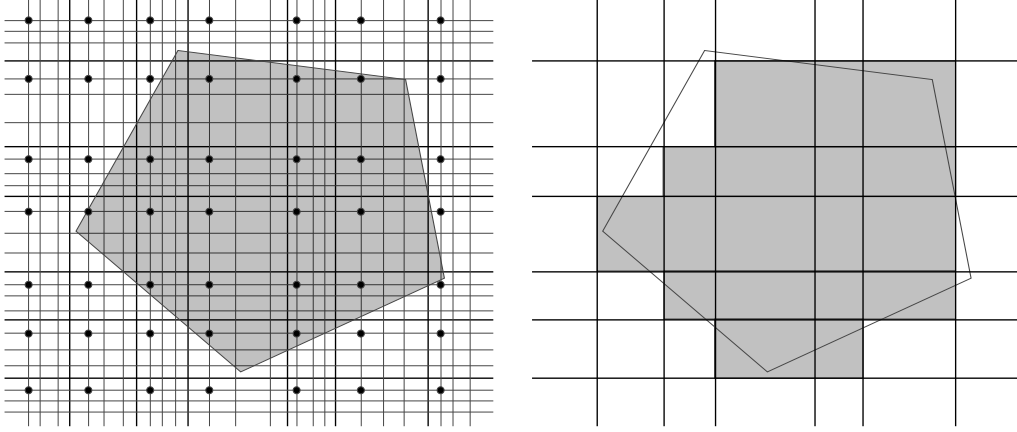
► **Corollary 1.11** (Informal, see full version). *Let $p : \mathbb{R}^d \rightarrow \mathbb{R}$ be a degree- k polynomial, and let $\epsilon > 0$. For $r \geq 3\sqrt{24}dk/\epsilon$ and any $r \times \dots \times r$ grid partition of \mathbb{R}^d , p takes value 0 in at most ϵr^d grid cells.*

1.4 Polynomial Regression

The second step of downsampling is to find a testing or learning algorithm that works for the uniform distribution over the (not necessarily evenly-spaced) hypergrid. Most of our learning results use *polynomial regression*. This is a powerful technique introduced in [30] that performs linear regression over a vector space of functions that approximately spans the hypothesis class. This method is usually applied by using Fourier analysis to construct such an approximate basis for the hypothesis class [10, 22, 15]. This was the method used, for example, by Blais, O’Donnell, & Wimmer [10] to achieve the $\text{poly}(dn)$ -time algorithms for intersections of halfspaces.

We take the same approach but we use the Walsh basis for functions on domain $[n]^d$ (see e.g. [11]) instead of the bases used in the prior works. We show that if one can establish bounds on the noise sensitivity in the Fourier basis for the hypothesis class restricted to the uniform distribution over $\{\pm 1\}^d$, then one gets a bound on the number of Walsh functions required to approximately span the “downsampled” hypothesis class. In this way, we establish that if one can apply standard Fourier-analytic techniques to the hypothesis class over the *uniform* distribution on $\{\pm 1\}^d$ and calculate the block boundary size, then the results for the hypercube essentially carry over to product distributions on \mathbb{R}^d .

An advantage of this technique is that both noise sensitivity and block boundary size grow at most linearly during function composition: for functions $f(x) = g(h_1(x), \dots, h_k(x))$ where each h_i belongs to the class \mathcal{H} , the noise sensitivity and block boundary size grow at most linearly in k . Therefore learning results for \mathcal{H} obtained in this way are easy to extend to arbitrary compositions of \mathcal{H} , which is how we get our result for intersections of halfspaces.



■ **Figure 1** Left: Random grid X (pale lines) with induced block partition (thick lines) and blockpoint values (dots), superimposed on $f^{-1}(1)$ (gray polygon). Right: f^{coarse} (grey) compared to f (polygon outline).

2 Downsampling

We will now introduce the main definitions, notation, and lemmas required by our main results. The purpose of this section is to establish the main conceptual component of the downsampling technique: that functions with small enough block boundary size can be efficiently well-approximated by a “coarsened” version of the function that is obtained by random sampling. See Figure 1 for an illustration of the following definitions.

▶ **Definition 2.1** (Block Partitions). *An r -block partition of \mathbb{R}^d is a pair of functions $\text{block} : \mathbb{R}^d \rightarrow [r]^d$ and $\text{blockpoint} : [r]^d \rightarrow \mathbb{R}^d$ obtained as follows. For each $i \in [d], j \in [r-1]$ let $a_{i,j} \in \mathbb{R}$ such that $a_{i,j} < a_{i,j+1}$ and define $a_{i,0} = -\infty, a_{i,r} = \infty$ for each i . For each $i \in [d], j \in [r]$ define the interval $B_{i,j} = (a_{i,j-1}, a_{i,j}]$ and a point $b_{i,j} \in B_{i,j}$. The function $\text{block} : \mathbb{R}^d \rightarrow [r]^d$ is defined by setting $\text{block}(x)$ to be the unique vector $v \in [r]^d$ such that $x_i \in B_{i,v_i}$ for each $i \in [d]$. The function $\text{blockpoint} : [r]^d \rightarrow \mathbb{R}^d$ is defined by setting $\text{blockpoint}(v) = (b_{1,v_1}, \dots, b_{d,v_d})$; note that $\text{blockpoint}(v) \in \text{block}^{-1}(v)$ where $\text{block}^{-1}(v) = \{x \in \mathbb{R}^d : \text{block}(x) = v\}$.*

▶ **Definition 2.2** (Block Functions and Coarse Functions). *For a function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$, we define $f^{\text{block}} : [r]^d \rightarrow \{\pm 1\}$ as $f^{\text{block}} := f \circ \text{blockpoint}$ and $f^{\text{coarse}} : \mathbb{R}^d \rightarrow \mathbb{R}$ as $f^{\text{coarse}} := f^{\text{block}} \circ \text{block} = f \circ \text{blockpoint} \circ \text{block}$. For any set \mathcal{H} of functions $\mathbb{R}^d \rightarrow \{\pm 1\}$, we define $\mathcal{H}^{\text{block}} := \{f^{\text{block}} \mid f \in \mathcal{H}\}$. For a distribution μ over \mathbb{R}^d and an r -block partition $\text{block} : \mathbb{R}^d \rightarrow [r]^d$ we define the distribution $\text{block}(\mu)$ over $[r]^d$ as the distribution of $\text{block}(x)$ for $x \sim \mu$.*

▶ **Definition 2.3** (Induced Block Partitions). *When μ is a product distribution over \mathbb{R}^d , a random grid X of length m is the grid obtained by sampling m points $x_1, \dots, x_m \in \mathbb{R}^d$ independently from μ and for each $i \in [d], j \in [m]$ defining $X_{i,j}$ to be the j^{th} -smallest coordinate in dimension i among all sampled points. For any r that divides m we define an r -block partition depending on X by defining for each $i \in [d], j \in [r-1]$ the point $a_{i,j} = X_{i,mj/r}$ so that the intervals are $B_{i,j} := (X_{i,m(j-1)/r}, X_{i,mj/r}]$ when $j \in \{2, \dots, r-1\}$ and $B_{i,1} = (-\infty, X_{i,m/r}]$, $B_{i,r} = (X_{i,m(r-1)/r}, \infty)$; we let the points $b_{i,j}$ defining blockpoint be arbitrary. This is the r -block partition induced by X .*

► **Definition 2.4** (Block Boundary Size). For a block partition $\mathbf{block} : \mathbb{R}^d \rightarrow [r]^d$, a distribution μ over \mathbb{R}^d , and a function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$, we say f is non-constant on a block $v \in [r]^d$ if there are sets $S, T \subset \mathbf{block}^{-1}(v)$ such that $\forall s \in S, t \in T : f(s) = 1, f(t) = -1$; and S, T have positive measure (in the product of Lebesgue measures). For a function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ and a number r , we define the r -block boundary size $\mathbf{bbs}(f, r)$ as the maximum number of blocks on which f is non-constant, where the maximum is taken over all r -block partitions $\mathbf{block} : \mathbb{R}^d \rightarrow [r]^d$. For a set \mathcal{H} of functions $\mathbb{R}^d \rightarrow \{\pm 1\}$, we define $\mathbf{bbs}(\mathcal{H}, r) := \max\{\mathbf{bbs}(f, r) \mid f \in \mathcal{H}\}$.

The total variation distance between two distributions μ, ν over a finite domain \mathcal{X} is defined as

$$\|\mu - \nu\|_{\text{TV}} := \frac{1}{2} \sum_{x \in \mathcal{X}} |\mu(x) - \nu(x)| = \max_{S \subseteq \mathcal{X}} |\mu(S) - \nu(S)|.$$

The essence of downsampling is apparent in the next proposition. It shows that the distance of f to its coarsened version f^{coarse} is bounded by two quantities: the fraction of blocks in the r -block partition on which f is not constant, and the distance of the distribution $\mathbf{block}(\mu)$ to uniform. When both quantities are small, testing or learning f can be done by testing or learning f^{coarse} instead. The uniform distribution over a set S is denoted $\text{unif}(S)$:

► **Proposition 2.5.** Let μ be a continuous product distribution over \mathbb{R}^d , let X be a random grid, and let $\mathbf{block} : \mathbb{R}^d \rightarrow [r]^d$ be the induced r -block partition. Then, for any measurable $f : \mathbb{R}^d \rightarrow \{\pm 1\}$, the following holds with probability 1 over the choice of X :

$$\mathbb{P}_{x \sim \mu} [f(x) \neq f^{\text{coarse}}(x)] \leq r^{-d} \cdot \mathbf{bbs}(f, r) + \|\mathbf{block}(\mu) - \text{unif}([r]^d)\|_{\text{TV}}.$$

Proof. We first establish that, with probability 1 over X and $x \sim \mu$, if $f(x) \neq f^{\text{coarse}}(x)$ then f is non-constant on $\mathbf{block}(x)$. Fix X and suppose there exists a set Z of positive measure such that for each $x \in Z$, $f(x) \neq f^{\text{coarse}}(x)$ but f is not non-constant on $\mathbf{block}(x)$, i.e. for $V = \mathbf{block}^{-1}(\mathbf{block}(x))$, either $\mu(V \cap f^{-1}(1)) = \mu(V)$ or $\mu(V \cap f^{-1}(-1)) = \mu(V)$. Then there is $v \in [r]^d$ such that for $V = \mathbf{block}^{-1}(v)$, $\mu(Z \cap V) > 0$. Let $y = \mathbf{blockpoint}(v)$. If $\mu(V \cap f^{-1}(f(y))) = \mu(V)$ then $\mu(Z \cap V) = 0$, so $\mu(V \cap f^{-1}(f(y))) = 0$. But for random X , the probability that there exists $v \in [r]^d$ such that $\mu(V \cap f^{-1}(\mathbf{blockpoint}(v))) = 0$ is 0, since $\mathbf{blockpoint}(v)$ is random within V .

Assuming that the above event occurs,

$$\begin{aligned} \mathbb{P}_{x \sim \mu} [f(x) \neq f^{\text{coarse}}(x)] &\leq \mathbb{P}_{x \sim \mu} [f \text{ is non-constant on } \mathbf{block}(x)] \\ &\leq \mathbb{P}_{v \sim [r]^d} [f \text{ is non-constant on } v] + \|\mathbf{block}(\mu) - \text{unif}([r]^d)\|_{\text{TV}}. \end{aligned}$$

Since $v \sim [r]^d$ is uniform, the probability of hitting a non-constant block is at most $r^{-d} \cdot \mathbf{bbs}(f, r)$. ◀

Next we give a bound on the number of samples required to ensure that $\mathbf{block}(\mu)$ is close to uniform. We need the following lemma.

► **Lemma 2.6.** Let μ be continuous probability distribution over \mathbb{R} , $m, r \in \mathbb{N}$ such that r divides m , and $\delta \in (0, 1/2)$. Let X be a set of m points sampled independently from μ . Write $X = \{x_1, \dots, x_m\}$ labeled such that $x_1 < \dots < x_m$ (and write $x_0 = -\infty$). Then for any $i \in [r]$,

$$\mathbb{P} \left[\mu \left(x_{(i-1)(m/r)}, x_{i(m/r)} \right) < \frac{1 - \delta}{r} \right] \leq 4 \cdot e^{-\frac{\delta^2 m}{32r}}.$$

71:12 Downsampling for Testing and Learning in Product Distributions

Proof. We assume that $i - 1 \leq r/2$. If $i - 1 > r/2$ then we can repeat the following analysis with the opposite ordering on the points in X . Write $x^* = x_{(i-1)\frac{m}{r}}$ and $\beta = \mu(-\infty, x^*]$. First suppose that $(1 - \delta/2)\frac{i-1}{r} < \beta < (1 + \delta/2)\frac{i-1}{r} \leq (1 + \delta/2)/2$; we will bound the probability of this event later.

Let $t \in \mathbb{R}$ be the point such that $\mu(x^*, t] = (1 - \delta)/r$ (which must exist since μ is continuous). Let $\eta = \frac{\delta}{1-\delta} \geq \delta$. Write $X^* = \{x \in X : x > x^*\}$. The expected value of $|X^* \cap (x^*, t]|$ is $|X^*| \frac{1-\delta}{r(1-\beta)} = (1 - \frac{i-1}{r}) \frac{1-\delta}{r(1-\beta)}$, where the factor $1 - \beta$ in the denominator is due to the fact that each element of X^* is sampled from μ conditional on being larger than x^* . The event $\mu(x^*, x_{i(m/r)}) < (1 - \delta)/r$ occurs if and only if $|X^* \cap (x^*, t]| > m/r$, which occurs with probability

$$\mathbb{P} \left[|X^* \cap (x^*, t]| > \frac{m}{r} \right] = \mathbb{P} \left[|X^* \cap (x^*, t]| > m \left(1 - \frac{(i-1)}{r} \right) \frac{1-\delta}{r(1-\beta)} (1 + \eta) \right]$$

where

$$\begin{aligned} 1 + \eta &= \frac{(1-\beta)}{(1-\delta)(1-\frac{i-1}{r})} \geq \frac{(1-(1+\delta/2)\frac{i-1}{r})}{(1-\delta)(1-\frac{i-1}{r})} = \frac{1}{1-\delta} \left(1 - \frac{(\delta/2)(i-1)}{r-(i-1)} \right) \\ &\geq \frac{1-\delta/2}{1-\delta} = 1 + \frac{\delta}{2(1-\delta)} \geq 1 + \delta/2. \end{aligned}$$

Since the expected value satisfies

$$|X^*| \frac{1-\delta}{r(1-\beta)} \geq \frac{m}{r} \left(1 - \frac{i-1}{r} \right) \frac{2(1-\delta)}{1-\delta/2} \geq \frac{m}{r} (1 - \delta/2) \geq \frac{m}{2r},$$

the Chernoff bound gives

$$\mathbb{P} \left[|X^* \cap (x^*, t]| > \frac{m}{r} \right] \leq \exp \left(-\frac{\delta^2 |X^*| (1-\delta)}{3 \cdot 4 \cdot r(1-\beta)} \right) \leq e^{-\frac{\delta^2 m}{3 \cdot 4 \cdot 2r}}.$$

Now let $t \in \mathbb{R}$ be the point such that $\mu(x^*, t] = (1 + \delta)/r$. The expected value of $|X^* \cap (x^*, t]|$ is now $|X^*| \frac{1+\delta}{r(1-\beta)}$. The event $\mu(x^*, x_{i(m/r)}) > (1 + \delta)/r$ occurs if and only if $|X^* \cap (x^*, t]| < m/r$, which occurs with probability

$$\mathbb{P} \left[|X^* \cap (x^*, t]| < \frac{m}{r} \right] = \mathbb{P} \left[|X^* \cap (x^*, t]| < m \left(1 - \frac{i-1}{r} \right) \frac{1+\delta}{r(1-\beta)} (1 - \eta) \right]$$

where

$$\begin{aligned} 1 - \eta &= \frac{1-\beta}{(1+\delta)(1-\frac{i-1}{r})} \leq \frac{1-(1+\delta/2)\frac{i-1}{r}}{(1+\delta)(1-\frac{i-1}{r})} = \frac{1}{1+\delta} \left(1 + \frac{(\delta/2)(i-1)}{r-(i-1)} \right) \\ &\leq \frac{1+\delta/2}{1+\delta} = 1 - \frac{\delta/2}{1+\delta} \leq 1 - \frac{\delta}{4}. \end{aligned}$$

The expected value satisfies $|X^*| \frac{1+\delta}{r(1-\beta)} > m/r$, so the Chernoff bound gives

$$\mathbb{P} \left[|X^* \cap (x^*, t]| < \frac{m}{r} \right] \leq \exp \left(-\frac{\delta^2 |X^*| (1+\delta)}{2 \cdot 4^2 \cdot r(1-\beta)} \right) \leq e^{-\frac{\delta^2 m}{2 \cdot 4^2}}.$$

It remains to bound the probability that $(1 - \delta/2)\frac{i-1}{r} < \beta < (1 + \delta/2)\frac{i-1}{r}$. Define $t \in \mathbb{R}$ such that $\mu(-\infty, t] = (1 + \delta/2)\frac{i-1}{r}$. $\beta = \mu(-\infty, x^*] \geq (1 + \delta/2)\frac{i-1}{r}$ if and only if $x^* > t$, i.e. $|X \cap (-\infty, t]| < \frac{i-1}{r}$. The expected value of $|X \cap (-\infty, t]|$ is $m \frac{(1+\delta/2)(i-1)}{r}$, so for $\eta = \frac{\delta/2}{1+\delta/2} \geq \delta/3$, the Chernoff bound implies

$$\begin{aligned} \mathbb{P} \left[|X \cap (-\infty, t]| < m \frac{i-1}{r} \right] &= \mathbb{P} \left[|X \cap (-\infty, t]| < m \frac{(1+\delta/2)(i-1)}{r} (1 - \eta) \right] \\ &\leq e^{-\frac{\delta^2 m (1+\delta/2)(i-1)}{18r}} \leq e^{-\frac{\delta^2 m}{18r}}. \end{aligned}$$

Now define $t \in \mathbb{R}$ such that $\mu(-\infty, t] = (1 - \delta/2) \frac{i-1}{r}$. $\beta = \mu(-\infty, x^*] \leq (1 - \delta/2) \frac{i-1}{r}$ if and only if $x^* < t$, i.e. $|X \cap (-\infty, t]| > \frac{i-1}{r}$. The expected value of $|X \cap (-\infty, t]|$ is $m \frac{(1-\delta/2)(i-1)}{r}$, so for $\eta = \frac{\delta}{2-\delta} \geq \delta/2$,

$$\begin{aligned} \mathbb{P} \left[|X \cap (-\infty, t]| > m \frac{i-1}{r} \right] &= \mathbb{P} \left[|X \cap (-\infty, t]| > m \frac{(1-\delta/2)(i-1)}{r} (1 + \eta) \right] \\ &\leq e^{-\frac{\delta^2 m (1-\delta/2)(i-1)}{2 \cdot 4r}} \leq e^{-\frac{\delta^2 m}{4^2 r}}. \end{aligned}$$

The conclusion then follows from the union bound over these four events. \blacktriangleleft

► **Lemma 2.7.** *Let $\mu = \mu_1 \times \dots \times \mu_d$ be a product distribution over \mathbb{R}^d where each μ_i is continuous. Let X be a random grid with length m sampled from μ , and let $\mathbf{block} : \mathbb{R}^d \rightarrow [r]^d$ be the r -block partition induced by X . Then*

$$\mathbb{P}_X \left[\|\mathbf{block}(\mu) - \mathbf{unif}([r]^d)\|_{\text{TV}} > \epsilon \right] \leq 4rd \cdot e^{-\frac{\epsilon^2 m}{18rd^2}}$$

Proof. For a fixed grid X and each $i \in [d]$, write $p_i : [r] \rightarrow [0, 1]$ be the probability distribution on $[r]$ with $p_i(z) = \mu_i(B_{i,z})$. Then $\mathbf{block}(\mu) = p_1 \times \dots \times p_d$.

Let $\delta = \frac{4\epsilon}{3d}$. Suppose that for every $i, j \in [d] \times [r]$ it holds that $\frac{1+\delta}{r} \leq p_i(j) \leq \frac{1-\delta}{r}$. Note that $d\delta = \frac{4\epsilon}{3} \leq \ln(1+2\epsilon) \leq 2\epsilon$. Then for every $v \in [r]^d$,

$$\mathbb{P}_{u \sim \mu} [\mathbf{block}(u) = v] = \prod_{i=1}^d p_i(v_i) \begin{cases} \leq (1+\delta)^d r^{-d} \leq e^{d\delta} r^{-d} \leq (1+2\epsilon) r^{-d} \\ \geq (1-\delta)^d r^{-d} \geq (1-d\delta) r^{-d} \geq (1-2\epsilon) r^{-d}. \end{cases}$$

So

$$\|\mathbf{block}(\mu) - \mathbf{unif}([r]^d)\|_{\text{TV}} = \frac{1}{2} \sum_{v \in [r]^d} \left| \mathbb{P}_{u \sim \mu} [\mathbf{block}(u) = v] - r^{-d} \right| \leq \frac{1}{2} \sum_{v \in [r]^d} 2\epsilon r^{-d} = \epsilon.$$

By Lemma 2.6 and the union bound, the probability that there is some $i \in [d], j \in [r]$ that satisfies $p_i(j) < (1 - \delta)/r$ is at most $4rd \cdot e^{-\frac{\epsilon^2 m}{18rd^2}}$. \blacktriangleleft

3 Testing Monotonicity

3.1 Testing Monotonicity on the Hypergrid

A good introduction to downsampling is the following short proof of the main result of Black, Chakrabarty, & Seshadhri [6]. In an earlier work, [5], they gave an $O((d^{5/6}/\epsilon^{4/3}) \text{poly} \log(dn))$ tester for the domain $[n]^d$, and in the later work they showed how to reduce the domain $[n]^d$ to $[r]^d$ for $r = \text{poly}(d/\epsilon)$.

Our monotonicity tester will use as a subroutine the following tester for *diagonal* functions. For a hypergrid $[n]^d$, a *diagonal* is a subset of points $\{x \in [n]^d : x = v + \lambda \vec{1}, \lambda \in \mathbb{Z}\}$ defined by some $v \in [n]^d$. A function $f : [n]^d \rightarrow \{0, 1\}$ is a *diagonal function* if it has at most one 1-valued point in each diagonal.

► **Lemma 3.1.** *There is an ϵ -tester with one-sided error and query complexity $O\left(\frac{1}{\epsilon} \log^2(1/\epsilon)\right)$ for diagonal functions on $[n]^d$.*

71:14 Downsampling for Testing and Learning in Product Distributions

Proof. For each $t \in [n]$ let D_t be the set of diagonals with length t . For any $x \in [n]^d$ let $\text{diag}(x)$ be the unique diagonal that contains x . For input $f : [n]^d \rightarrow \{0, 1\}$ and any $x \in [n]^d$, let $R(x) = \frac{|\{y \in \text{diag}(x) : f(y) = 1\}|}{|\text{diag}(x)|}$.

Suppose that f is ϵ -far from diagonal. Then f must have at least ϵn^d 1-valued points; otherwise we could set each 1-valued point to 0 to obtain the constant 0 function. Now observe

$$\begin{aligned} \mathbb{E}_{x \sim [n]^d} [R(x)] &= \mathbb{E}_{x \sim [n]^d} \left[\sum_{t=1}^n \sum_{L \in D_t} \mathbf{1}[\text{diag}(x) = L] \frac{|\{y \in L : f(y) = 1\}|}{t} \right] \\ &= \sum_{t=1}^n \sum_{L \in D_t} \mathbb{P}_{x \sim [n]^d} [x \in L] \frac{|\{y \in L : f(y) = 1\}|}{t} = \sum_{t=1}^n \sum_{L \in D_t} \frac{t}{n^d} \frac{|\{y \in L : f(y) = 1\}|}{t} \\ &= \frac{1}{n^d} |\{y \in [n]^d : f(y) = 1\}| \geq \epsilon. \end{aligned}$$

For each i , define $A_i = \{x \in [n]^d : \frac{1}{2^i} < R(x) \leq \frac{1}{2^{i-1}}\}$. Let $k = \log(4/\epsilon)$. Then

$$\begin{aligned} \epsilon &\leq \mathbb{E} [R(x)] \leq \sum_{i=1}^{\infty} \frac{|A_i|}{n^d} \max_{x \in A_i} R(x) \leq \sum_{i=1}^{\infty} \frac{|A_i|}{n^d 2^{i-1}} \leq \sum_{i=1}^k \frac{|A_i|}{n^d 2^{i-1}} + \sum_{i=k+1}^{\infty} \frac{1}{2^{i-1}} \\ &\leq \sum_{i=1}^k \frac{|A_i|}{n^d 2^{i-1}} + \frac{1}{2^{k-1}} \leq \sum_{i=1}^k \frac{|A_i|}{n^d 2^{i-1}} + \frac{\epsilon}{2} \\ \implies \frac{\epsilon}{2} &\leq \sum_{i=1}^k \frac{|A_i|}{n^d 2^{i-1}}. \end{aligned}$$

Therefore there is some $\ell \in [k]$ such that $|A_\ell| \geq \frac{\epsilon n^d 2^{\ell-1}}{2k}$.

The tester is as follows. For each $i \in [k]$:

1. Sample $p = \frac{k}{\epsilon 2^{i-2}} \ln(6)$ points $x_1, \dots, x_p \sim [n]^d$.
2. For each $j \in [p]$, sample $q = 2^{i+2} \ln(12)$ points y_1, \dots, y_q from $\text{diag}(x_j)$ and reject if there are two distinct 1-valued points in the sample.

The query complexity of the tester is $\sum_{i=1}^k 4^2 \ln(6) \ln(12) \frac{k}{\epsilon 2^i} 2^i = O\left(\frac{1}{\epsilon} \log^2(1/\epsilon)\right)$.

The tester will clearly accept any diagonal function. Now suppose that f is ϵ -far from having this property, and let $\ell \in [k]$ be such that $|A_\ell| \geq \frac{\epsilon n^d 2^{\ell-2}}{k}$. On iteration $i = \ell$, the algorithm samples $p = \frac{k}{\epsilon 2^{\ell-2}} \ln(6)$ points x_1, \dots, x_p . The probability that $\forall j \in [p], x_j \notin A_\ell$ is at most

$$\left(1 - \frac{|A_\ell|}{n^d}\right)^p \leq \left(1 - \frac{\epsilon 2^{\ell-2}}{k}\right)^p \leq \exp\left(-\frac{\epsilon p 2^{\ell-2}}{k}\right) \leq 1/6.$$

Now assume that there is some $x_j \in A_\ell$, so that $R(x_j) > 2^{-\ell}$. Let $A, B \subset \text{diag}(x_j)$ be disjoint subsets that partition the 1-valued points in $\text{diag}(x_j)$ into equally-sized parts. Then for y sampled uniformly at random from $\text{diag}(x_j)$, $\mathbb{P}[y \in A], \mathbb{P}[y \in B] \geq 2^{-(\ell+1)}$. The probability that there are at least 2 distinct 1-valued points in y_1, \dots, y_q sampled by the algorithm is at least the probability that one of the first $q/2$ samples is in A and one of the last $q/2$ samples is in B . This fails to occur with probability at most $2(1 - 2^{-(\ell+1)})^{q/2} \leq 2e^{-q 2^{-(\ell+2)}} \leq 1/6$. So the total probability of failure is at most $2/6 = 1/3$. \blacktriangleleft

► Theorem 3.2. *There is a non-adaptive monotonicity tester on domain $[n]^d$ with one-sided error and query complexity $\tilde{O}\left(\frac{d^{5/6}}{\epsilon^{4/3}}\right)$.*

Proof. Set $r = \lceil 4d/\epsilon \rceil$, and assume without loss of generality that r divides n . Partition $[n]$ into r intervals $B_i = \{(i-1)(n/r) + 1, \dots, i(n/r)\}$. For each $v \in [r]^d$ write $B_v = B_{v_1} \times \dots \times B_{v_d}$. Define $\mathbf{block} : [n]^d \rightarrow [r]^d$ where $\mathbf{block}(x)$ is the unique vector $v \in [r]^d$ such that $x \in B_v$. Define $\mathbf{block}^{-\downarrow}(v) = \min\{x \in B_v\}$ and $\mathbf{block}^{-\uparrow}(v) = \max\{x \in B_v\}$, where the minimum and maximum are with respect to the natural ordering on $[n]^d$. For $f : [n]^d \rightarrow \{0, 1\}$, write $f^{\mathbf{block}} : [r]^d \rightarrow \{0, 1\}$, $f^{\mathbf{block}}(v) = f(\mathbf{block}^{-\downarrow}(v))$. We may simulate queries v to $f^{\mathbf{block}}$ by returning $f(\mathbf{block}^{-\downarrow}(v))$. We will call $v \in [r]^d$ a *boundary block* if $f(\mathbf{block}^{-\downarrow}(v)) \neq f(\mathbf{block}^{-\uparrow}(v))$.

The test proceeds as follows: On input $f : [n]^d \rightarrow \{0, 1\}$ and a block $v \in [r]^d$, define the following functions:

$$g : [n]^d \rightarrow \{0, 1\}, \quad g(x) = \begin{cases} f^{\mathbf{block}}(\mathbf{block}(x)) & \text{if } \mathbf{block}(x) \text{ is not a boundary block} \\ f(x) & \text{if } \mathbf{block}(x) \text{ is a boundary block.} \end{cases}$$

$$b : [r]^d \rightarrow \{0, 1\}, \quad b(v) = \begin{cases} 0 & \text{if } v \text{ is not a boundary block} \\ 1 & \text{if } v \text{ is a boundary block.} \end{cases}$$

$$h : [r]^d \rightarrow \{0, 1\}, \quad h(v) = \begin{cases} f^{\mathbf{block}}(v) & \text{if } v \text{ is not a boundary block} \\ 0 & \text{if } v \text{ is a boundary block.} \end{cases}$$

Queries to each of these functions can be simulated by 2 or 3 queries to f . The tester performs:

1. Test whether $g = f$, or whether $\text{dist}(f, g) > \epsilon/4$, using $O(1/\epsilon)$ queries.
2. Test whether b is diagonal, or is $\epsilon/4$ -far from diagonal, using Lemma 3.1, with $O(\frac{1}{\epsilon} \log^2(1/\epsilon))$ queries.
3. Test whether h is monotone or $\epsilon/4$ -far from monotone, using the tester of Black, Chakrabarty, & Seshadhri with $\tilde{O}\left(\frac{d^{5/6}}{\epsilon^{4/3}}\right)$ queries.

▷ **Claim 3.3.** If f is monotone, the tester passes all 3 tests with probability 1.

Proof of claim. To see that $g = f$, observe that if $v = \mathbf{block}(x)$ is not a boundary block then $f(\mathbf{block}^{-\downarrow}(v)) = f(\mathbf{block}^{-\uparrow}(v))$. If $f(x) \neq f^{\mathbf{block}}(\mathbf{block}(x))$ then $f(x) \neq f(\mathbf{block}^{-\downarrow}(v))$ and $f(x) \neq f(\mathbf{block}^{-\uparrow}(v))$ while $\mathbf{block}^{-\downarrow}(v) \preceq x \preceq \mathbf{block}^{-\uparrow}(v)$, and this is a violation of the monotonicity of f . Therefore f will pass the first test with probability 1.

To see that f passes the second test with probability 1, observe that if f had 2 boundary blocks in some diagonal, then there are boundary blocks $u, v \in [r]^d$ such that $\mathbf{block}^{-\uparrow}(u) \prec \mathbf{block}^{-\downarrow}(v)$. But then there is $x, y \in [n]^d$ such that $\mathbf{block}(x) = u, \mathbf{block}(y) = v$ and $f(x) = 1, f(y) = 0$; since $x \preceq \mathbf{block}^{-\uparrow}(u) \prec \mathbf{block}^{-\downarrow}(v) \preceq y$, this contradicts the monotonicity of f . So f has at most 1 boundary block in each diagonal.

To see that h is monotone, it is sufficient to consider the boundary blocks, since all other values are the same as $f^{\mathbf{block}}$. Let $v \in [r]^d$ be a boundary block, so there exist $x, y \in [n]^d$ such that $\mathbf{block}(x) = \mathbf{block}(y)$ and $f(x) = 1, f(y) = 0$. Suppose $u \prec v$ is not a boundary block (if it is a boundary block then $h(u) = h(v) = 0$). If $h(u) = 1$ then $f(\mathbf{block}^{-\downarrow}(u)) = 1$, but $\mathbf{block}^{-\downarrow}(u) \prec \mathbf{block}^{-\downarrow}(v) \preceq y$ while $f(\mathbf{block}^{-\downarrow}(u)) > f(y)$, a contradiction. So it must be that $h(u) = 0$ whenever $u \prec v$. For any block $u \in [r]^d$ such that $v \prec u$, we have $0 = h(v) \leq h(u)$, so monotonicity holds. Since the tester of Black, Chakrabarty, & Seshadhri has one-sided error, the test passes with probability 1. ◁

▷ **Claim 3.4.** If g is $\epsilon/4$ -close to f , b is $\epsilon/4$ -close to diagonal, and h is $\epsilon/4$ -close to monotone, then f is ϵ -close to monotone.

71:16 Downsampling for Testing and Learning in Product Distributions

Proof of claim. Let $h^{\text{coarse}} : [n]^d \rightarrow \{0, 1\}$ be the function $h^{\text{coarse}}(x) = h(\text{block}(x))$. Suppose that $f(x) \neq h^{\text{coarse}}(x)$. If $v = \text{block}(x)$ is not a boundary block of f then $h^{\text{coarse}}(x) = h(v) = f^{\text{block}}(v) = g(x)$, so $f(x) \neq g(x)$. If v is a boundary block then $h^{\text{coarse}}(x) = h(v) = 0$ so $f(x) = 1$, and $b(v) = 1$.

Suppose for contradiction that there are more than $\frac{\epsilon}{2}r^d$ boundary blocks $v \in [r]^d$, so there are more than $\frac{\epsilon}{2}r^d$ 1-valued points of b . Any diagonal function has at most dr^{d-1} 1-valued points. Therefore the distance of b to diagonal is at least

$$r^{-d} \left(\frac{\epsilon}{2}r^d - dr^{d-1} \right) = \frac{\epsilon}{2} - \frac{d}{r} = \frac{\epsilon}{2} - \frac{\epsilon}{4} = \frac{\epsilon}{4},$$

a contradiction. So f has at most $\frac{\epsilon}{2}r^d$ boundary blocks. Now

$$\text{dist}(f, h^{\text{coarse}}) = \text{dist}(f, g) + \mathbb{P}_{x \sim [n]^d} [f(x) = 1, \text{block}(x) \text{ is a boundary block}] \leq \frac{\epsilon}{4} + r^{-d} \cdot \frac{\epsilon r^d}{2} = \frac{3}{4}\epsilon.$$

Let $p : [r]^d \rightarrow \{0, 1\}$ be a monotone function minimizing the distance to h , and let $p^{\text{coarse}} : [n]^d \rightarrow \{0, 1\}$ be the function $p^{\text{coarse}}(x) = p(\text{block}(x))$. Then

$$\text{dist}(h^{\text{coarse}}, p^{\text{coarse}}) = \mathbb{P}_{x \sim [n]^d} [h(\text{block}(x)) \neq p(\text{block}(x))] = \mathbb{P}_{v \sim [r]^d} [h(v) \neq p(v)] \leq \epsilon/4.$$

Finally, the distance of f to the nearest monotone function is at most

$$\text{dist}(f, p^{\text{coarse}}) \leq \text{dist}(f, h^{\text{coarse}}) + \text{dist}(h^{\text{coarse}}, p^{\text{coarse}}) \leq \frac{3}{4}\epsilon + \frac{1}{4}\epsilon = \epsilon. \quad \triangleleft$$

These two claims suffice to establish the theorem. ◀

3.2 Monotonicity Testing for Product Distributions

The previous section used a special case of downsampling, tailored for the uniform distribution over $[n]^d$. We will call a product distribution $\mu = \mu_1 \times \dots \times \mu_d$ over \mathbb{R}^d *continuous* if each of its factors μ_i are continuous (i.e. absolutely continuous with respect to the Lebesgue measure). The proof for discrete distributions is in the full version.

► **Theorem 1.1.** *There is a one-sided, non-adaptive ϵ -tester for monotonicity of functions $\mathbb{R}^d \rightarrow \{0, 1\}$ that is distribution-free under (finite or continuous) product distributions, using*

$$O\left(\frac{d^{5/6}}{\epsilon^{4/3}} \text{poly} \log(d/\epsilon)\right)$$

queries and $O(\frac{d^3}{\epsilon^3} \log(d/\epsilon))$ samples.

Proof. We follow the proof of Theorem 3.2, with some small changes. Let $r = \lceil 16d/\epsilon \rceil$. The tester first samples a grid X with length $m = O\left(\frac{rd^2}{\epsilon^2} \log(rd)\right)$ and constructs the induced $(r+2)$ -block partition, with cells labeled $\{0, \dots, r+1\}^d$. We call a block $v \in \{0, \dots, r+1\}^d$ *upper extreme* if there is some $i \in [d]$ such that $v_i = r+1$, and we call it *lower extreme* if there is some $i \in [d]$ such that $v_i = 0$ but v is not upper extreme. Call the upper extreme blocks U and the lower extreme blocks L . Note that $[r]^d = \{0, \dots, r+1\}^d \setminus (U \cup L)$.

For each $v \in [r]^d$, we again define $\text{block}^{-\uparrow}(v), \text{block}^{-\downarrow}(v)$ as, respectively, the supremal and infimal point $x \in \mathbb{R}^d$ such that $\text{block}(x) = v$. The algorithm will ignore the extreme blocks $U \cup L$, which do not have a supremal or an infimal point. Therefore it is not defined whether these blocks are boundary blocks.

By Lemma 2.7, with probability at least $5/6$, we will have $\|\text{block}(\mu) - \text{unif}(\{0, \dots, r+1\})\|_{\text{TV}} \leq \epsilon/8$. We define b, h as before, with domain $[r]^d$. Define g similarly but with domain \mathbb{R}^d and values

$$g(x) = \begin{cases} 1 & \text{if } \text{block}(x) \in U \\ 0 & \text{if } \text{block}(x) \in L \\ f(x) & \text{if } \text{block}(x) \in [n]^d \text{ is a boundary block} \\ f^{\text{block}}(\text{block}(x)) & \text{otherwise.} \end{cases}$$

If f is monotone, it may now be the case $f \neq g$, but we will have $f(x) = g(x)$ for all x with $\text{block}(x) \in [r]^d$, where the algorithm will make its queries. The algorithm will test whether $f(x) = g(x)$ on all x with $\text{block}(x) \in [r]^d$, or $\epsilon/8$ -far from this property, which can be again done with $O(1/\epsilon)$ samples. Note that if f is $\epsilon/8$ -close to having this property, then

$$\begin{aligned} \text{dist}_\mu(f, g) &\leq \mathbb{P}_{x \sim \mu} [\text{block}(x) \notin [n]^d] + \epsilon/8 \\ &\leq \frac{d(r+2)^{d-1}}{(r+2)^d} + \epsilon/8 + \|\text{block}(\mu) - \text{unif}([r]^d \cup U \cup L)\|_{\text{TV}} \\ &\leq \frac{\epsilon}{16} + \frac{\epsilon}{8} + \frac{\epsilon}{4} \leq \frac{\epsilon}{2}. \end{aligned}$$

The algorithm then proceeds as before, with error parameter $\epsilon/2$. To test whether $g = f$, the algorithm samples from μ and throws away any sample $x \in \mathbb{R}^d$ with $\text{block}(x) \notin [r]^d$. It then tests b and h using the uniform distribution on $[r]^d$. It suffices to prove the following claim, which replaces Claim 3.4.

▷ **Claim 3.5.** If g is $\epsilon/2$ -close to f , b is $\epsilon/16$ -close to diagonal, and h is $\epsilon/8$ -close to monotone, then f is ϵ -close to monotone.

Proof of claim. Let $p : [r]^d \rightarrow \{0, 1\}$ be a monotone function minimizing the distance to h . Then $p(v) \neq h(v)$ on at most $\frac{\epsilon r^d}{8}$ blocks $v \in [r]^d$. Define $p^{\text{coarse}} : \mathbb{R}^d \rightarrow \{0, 1\}$ as $p^{\text{coarse}}(x) = p(\text{block}(x))$ when $\text{block}(x) \in [r]^d$, and $p^{\text{coarse}}(x) = g(x)$ when $\text{block}(x) \in U \cup L$. Note that p^{coarse} is monotone.

By the triangle inequality,

$$\text{dist}_\mu(f, p^{\text{coarse}}) \leq \text{dist}_\mu(f, g) + \text{dist}_\mu(g, p^{\text{coarse}}).$$

From above, we know $\text{dist}_\mu(f, g) \leq \epsilon/2$. To bound the second term, observe that since b is $\epsilon/16$ -close to diagonal, there are at most

$$\frac{\epsilon}{16} r^d + d r^{d-1} \leq \frac{\epsilon}{16} r^d + \frac{d}{r} r^d \leq \frac{\epsilon}{16} r^d + \frac{\epsilon}{16} r^d = \frac{\epsilon}{8} r^d$$

boundary blocks. Then observe that if $g(x) \neq p^{\text{coarse}}(x)$ then $\text{block}(x) \in [r]^d$ and either $\text{block}(x)$ is a boundary block, or $g(x) = f^{\text{block}}(\text{block}(x)) = h(\text{block}(x))$ and $h(\text{block}(x)) \neq p(\text{block}(x))$. Then

$$\begin{aligned} \text{dist}_\mu(g, p^{\text{coarse}}) &\leq \left(\frac{1}{(r+2)^d} \sum_{v \in [r]^d} \mathbf{1}[v \text{ is a boundary block, or } h(v) \neq p(v)] \right) \\ &\quad + \|\text{block}(\mu) - \text{unif}(\{0, \dots, r+1\}^d)\|_{\text{TV}} \\ &\leq \frac{\epsilon r^d}{8 r^d} + \frac{\epsilon r^d}{8 r^d} + \frac{\epsilon}{4} \leq \frac{\epsilon}{2}. \end{aligned}$$

◁

◀

References

- 1 Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Information and Computation*, 204(11):1704–1717, 2006.
- 2 Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 21–30, 2012.
- 3 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Tolerant testers of image properties. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- 4 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. *Random Structures & Algorithms*, 54(3):413–443, 2019.
- 5 Hadley Black, Deeparnab Chakrabarty, and C Seshadhri. A $o(d) \cdot \text{poly} \log n$ monotonicity tester for boolean functions over the hypergrid $[n]^d$. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2133–2151, 2018.
- 6 Hadley Black, Deeparnab Chakrabarty, and C Seshadhri. Domain reduction for monotonicity testing: A $o(d)$ tester for boolean functions in d -dimensions. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1975–1994, 2020.
- 7 Eric Blais and Abhinav Bommireddi. On testing and robust characterizations of convexity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 8 Eric Blais, Clément Canonne, Igor Oliveira, Rocco Servedio, and Li-Yang Tan. Learning circuits with few negations. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, page 512, 2015.
- 9 Eric Blais, Renato Ferreira Pinto Jr, and Nathaniel Harms. VC dimension and distribution-free sample-based testing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 504–517, 2021.
- 10 Eric Blais, Ryan O’Donnell, and Karl Wimmer. Polynomial regression under arbitrary product distributions. *Machine Learning*, 80(2-3):273–294, 2010.
- 11 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *Proceedings of the IEEE 29th Conference on Computational Complexity (CCC)*, pages 309–320, 2014.
- 12 Eric Blais and Yuichi Yoshida. A characterization of constant-sample testable properties. *Random Structures & Algorithms*, 55(1):73–88, 2019.
- 13 Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–127, 1992.
- 14 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.
- 15 Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k -monotonicity: The rise and fall of boolean functions. *Theory of Computing*, 15(1):1–55, 2019.
- 16 Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. *ACM Transactions on Algorithms (TALG)*, 13(2):1–30, 2017.
- 17 Deeparnab Chakrabarty and Comandur Seshadhri. An $o(n)$ monotonicity tester for boolean functions over the hypercube. *SIAM Journal on Computing*, 45(2):461–472, 2016.
- 18 Xi Chen, Adam Freilich, Rocco A Servedio, and Timothy Sun. Sample-based high-dimensional convexity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- 19 Xue Chen, Anindya De, and Rocco A Servedio. Testing noisy linear functions for sparsity. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 610–623, 2020.

- 20 Mónika Csikós, Nabil H Mustafa, and Andrey Kupavskii. Tight lower bounds on the VC-dimension of geometric set systems. *Journal of Machine Learning Research*, 20(81):1–8, 2019.
- 21 Anindya De, Elchanan Mossel, and Joe Neeman. Is your function low-dimensional? In *Conference on Learning Theory*, pages 979–993, 2019.
- 22 Ilias Diakonikolas, Prahladh Harsha, Adam Klivans, Raghu Meka, Prasad Raghavendra, Rocco A Servedio, and Li-Yang Tan. Bounding the average sensitivity and noise sensitivity of polynomial threshold functions. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 533–542, 2010.
- 23 Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1061–1073, 2018.
- 24 Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Transactions on Algorithms (TALG)*, 6(3):1–37, 2010.
- 25 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the 34th annual ACM symposium on Theory of Computing (STOC)*, pages 474–483, 2002.
- 26 Noah Fleming and Yuichi Yoshida. Distribution-free testing of linear functions on \mathbb{R}^n . In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 27 Oded Goldreich and Dana Ron. On sample-based testers. *ACM Transactions on Computation Theory (TOCT)*, 8(2):1–54, 2016.
- 28 Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM Journal on Computing*, 37(4):1107–1138, 2007.
- 29 Nathaniel Harms. Testing halfspaces over rotation-invariant distributions. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 694–713, 2019.
- 30 Adam T. Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.
- 31 Michael Kearns and Dana Ron. Testing problems with sublearning sample complexity. *Journal of Computer and System Sciences*, 61(3):428–456, 2000.
- 32 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric-type theorems. *SIAM Journal on Computing*, 47(6):2238–2276, 2018.
- 33 Adam R Klivans, Ryan O’Donnell, and Rocco A Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004.
- 34 Adam R Klivans, Ryan O’Donnell, and Rocco A Servedio. Learning geometric concepts via gaussian surface area. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 541–550, 2008.
- 35 Luis Rademacher and Santosh Vempala. Testing geometric convexity. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 469–480. Springer, 2004.
- 36 Sofya Raskhodnikova. Approximate testing of visual properties. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 370–381. Springer, 2003.
- 37 Dana Ron and Asaf Rosin. Optimal distribution-free sample-based testing of subsequence-freeness. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 337–256. SIAM, 2021.
- 38 Santosh Vempala. Learning convex concepts from gaussian distributions with PCA. In *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 124–130, 2010.
- 39 Santosh Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *Journal of the ACM*, 57(6):1–14, 2010.
- 40 Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.