

A Polynomial Lower Bound for Testing Monotonicity

Aleksandrs Belovs
Centrum Wiskunde &
Informatica
Science Park 123
Amsterdam, the Netherlands
stiboh@gmail.com

Eric Blais
David R. Cheriton School of
Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Canada
eric.blais@uwaterloo.ca

ABSTRACT

We show that every algorithm for testing n -variate Boolean functions for monotonicity has query complexity $\tilde{\Omega}(n^{1/4})$. All previous lower bounds for this problem were designed for non-adaptive algorithms and, as a result, the best previous lower bound for general (possibly adaptive) monotonicity testers was only $\Omega(\log n)$. Combined with the query complexity of the non-adaptive monotonicity tester of Khot, Minzer, and Safra (FOCS 2015), our lower bound shows that adaptivity can result in at most a quadratic reduction in the query complexity for testing monotonicity.

By contrast, we show that there is an exponential gap between the query complexity of adaptive and non-adaptive algorithms for testing regular linear threshold functions (LTFs) for monotonicity. Chen, De, Servedio, and Tan (STOC 2015) recently showed that non-adaptive algorithms require almost $\Omega(n^{1/2})$ queries for this task. We introduce a new adaptive monotonicity testing algorithm which has query complexity $O(\log n)$ when the input is a regular LTF.

Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—Computations on discrete structures

General Terms

Theory

Keywords

Property Testing, Adaptivity of query algorithms, Talagrand's Random DNF

1. INTRODUCTION

The Boolean function $f: \{0, 1\}^n \rightarrow \{0, 1\}$ is *monotone* iff $f(x) \leq f(y)$ for all $x \preceq y$, where \preceq is the bitwise partial order on the Boolean hypercube $\{0, 1\}^n$ (i.e., $x \preceq y$ iff $x_i \leq y_i$

for all $i \in [n]$). Conversely, the function f is ϵ -far from monotone for some $\epsilon > 0$ if for every monotone function $g: \{0, 1\}^n \rightarrow \{0, 1\}$, there are at least $\epsilon 2^n$ points $x \in \{0, 1\}^n$ such that $f(x) \neq g(x)$. An ϵ -tester for monotonicity is a bounded-error randomized algorithm that distinguishes monotone functions from those that are ϵ -far from monotone. The tester has oracle access to the function f . It is *non-adaptive* if its queries do not depend on the oracle's responses to the previous queries; otherwise, it is *adaptive*.

The study of the monotonicity testing problem was initiated in 1998 by Goldreich, Goldwasser, Lehman, and Ron [15], who introduced the natural *edge tester* for monotonicity. This tester selects edges $x \preceq y$ of the hypercube $\{0, 1\}^n$ uniformly at random and verifies that $f(x) \leq f(y)$ on each of these edges. Dodis, Goldreich, Lehman, Raskhodnikova, Ron, and Samorodnitsky [12] showed that this tester has query complexity $O(n/\epsilon)$. Goldreich *et al.* [16], in the journal version of [15], showed that the analysis of this algorithm is tight and asked: are there any other ϵ -testers for monotonicity with significantly smaller query complexity?

1.1 Previous Work on Monotonicity Testing

In 2002, Fischer *et al.* [14] showed that every non-adaptive tester for monotonicity has query complexity $\Omega(\log n)$.¹ This immediately implies an $\Omega(\log \log n)$ lower bound for the more general class of adaptive testers for monotonicity. Stronger lower bounds were established for more restricted classes of algorithms, like 1-sided non-adaptive algorithms [14] and even more limited *pair testers* [5]—algorithms that select pairs $x \preceq y$ of inputs from some distribution over the comparable pairs of inputs in the hypercube and check that $f(x) \leq f(y)$ on each selected pair. Algorithms and strong lower bounds were also introduced for the related problem of testing monotonicity of functions with non-Boolean ranges and other domains [12, 3, 8]. However, there was no further progress on Goldreich *et al.*'s original question for more than a decade, until a recent outburst of activity.

In 2013, Chakrabarty and Seshadhri [7] showed that there are indeed testers for monotonicity with query complexity asymptotically smaller than that of the edge tester. They introduced a pair tester with query complexity $\tilde{O}(n^{7/8} \epsilon^{-3/2})$. Chen, Servedio and Tan [10] further developed these ideas to obtain a pair tester with query complexity $\tilde{O}(n^{5/6} \epsilon^{-4})$. Khot, Minzer, and Safra [17] showed that a directed version of Talagrand's isoperimetric inequality yields a pair tester with query complexity $\tilde{O}(\sqrt{n}/\epsilon^2)$. The authors [1] used this

¹Throughout the paper, we assume that $\epsilon = \Theta(1)$ in the lower bound settings.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

STOC'16, June 19–21, 2016, Cambridge, MA, USA
ACM. 978-1-4503-4132-5/16/06...\$15.00
<http://dx.doi.org/10.1145/2897518.2897567>

inequality to develop a *quantum* tester for monotonicity with query complexity $\tilde{O}(n^{1/4}\epsilon^{-1/2})$.

On the lower bound side, Chen, Servedio and Tan [10] established a lower bound of $\tilde{\Omega}(n^{1/5})$ queries for all non-adaptive testers for monotonicity. This lower bound was later improved to almost $\Omega(\sqrt{n})$ by Chen, De, Servedio and Tan [9]. These recent developments essentially give a complete answer to the question of Goldreich *et al.* for non-adaptive algorithms: there exists a non-adaptive tester for monotonicity with query complexity that is quadratically smaller than that of the edge tester, and this gap is best possible.

1.2 Our Results

Despite all the recent progress on monotonicity, our understanding of the query complexity of *adaptive* testers for monotonicity remains far from complete. The best lower bound for the problem is $\Omega(\log n)$, which follows directly from the non-adaptive lower bound of Chen *et al.* [10]. This lower bound leaves open the possibility that there exist testers for monotonicity with query complexity that is *exponentially* smaller than that of the edge tester or of any other non-adaptive tester for monotonicity. Our main result eliminates this possibility.

THEOREM 1. *There exists an absolute constant $\epsilon > 0$ such that any (adaptive) randomized algorithm that ϵ -tests whether an n -variate Boolean function f is monotone makes $\tilde{\Omega}(n^{1/4})$ queries to f .*

Theorem 1 shows that the query complexity of any tester for monotonicity (adaptive or not) is at most a quartic factor better than that of the edge tester, and that adaptivity can result in at most a quadratic reduction in the query complexity for the monotonicity testing problem.

The proof of Theorem 1 is established by considering random functions known as *Talagrand's random DNFs*. These monotone functions have previously appeared in many different contexts—including DNF approximation [22], hardness amplification [4], and learning theory [18]—and are of particular interest because of their extremal noise sensitivity properties [20]. We use the same noise sensitivity properties to show that Talagrand's random DNF with \sqrt{n} random input variables negated is $\Omega(1)$ -far from monotone with high probability, and that a randomized algorithm with small query complexity cannot reliably distinguish original Talagrand's random DNFs from this modified version.

Our approach represents a notable departure from previous lower bounds for the monotonicity testing problem, in that all the previous lower bounds [14, 10, 9] were obtained by considering linear threshold functions (LTFs)—Boolean functions of the form $f(x) = \text{sgn}(\sum_{i \in [n]} w_i x_i - \theta)$ with appropriate weight $w_1, \dots, w_n \in \mathbb{R}$ and threshold $\theta \in \mathbb{R}$ parameters. In fact, the previous lower bounds for monotonicity testing were obtained by considering a special class of LTFs known as *regular LTFs*. An LTF is τ -regular when the magnitude of each weight w_i is bounded by $|w_i| \leq \tau \cdot \sqrt{\sum_{j \in [n]} w_j^2}$. Regular LTFs have been studied in the context of approximating [11], learning [21], and testing [19] LTFs; the lower bounds in [10, 9] are obtained by showing that non-adaptive algorithm with small query complexity cannot reliably distinguish $O(\frac{1}{\sqrt{n}})$ -regular LTFs that are monotone from those that are far from monotone.

Chen, De, Servedio, and Tan [9] asked if their approach could be generalized to obtain polynomial lower bounds on the query complexity of adaptive testers for monotonicity. We answer this question in the negative, by showing that there does exist an adaptive algorithm with logarithmic query complexity that can ϵ -test monotonicity when its input is promised to be a regular LTF.

THEOREM 2. *Fix $\epsilon > 0$ and $\tau > 0$. There is an adaptive algorithm \mathcal{A} with query complexity² $O_{\epsilon, \tau}(1) + \log n$ that, given oracle access to the n -variate Boolean function f ,*

1. *Always accepts when f is a monotone $\frac{\tau}{\sqrt{n}}$ -regular LTF,*
2. *and Rejects with probability at least $\frac{1}{2}$ when f is a $\frac{\tau}{\sqrt{n}}$ -regular LTF that is ϵ -far from monotone.*

Combined with the lower bound of Chen *et al.* [9], Theorem 2 shows that there are natural classes of functions for which adaptivity can reduce the query complexity of monotonicity testers by an exponential amount. By the standard reduction between adaptive and non-adaptive algorithms, this is best possible.

The proof of Theorem 2 is obtained by analyzing a natural adaptive tester for monotonicity. The tester selects pairs of inputs $x, y \in \{0, 1\}^n$ independently at random until it finds a pair for which $f(x) \neq f(y)$, then it performs a random binary search between x and y to identify an edge (z, z') of the hypercube on which $f(z) \neq f(z')$. It accepts if and only if f is monotone on this edge. This algorithm was first considered and communicated to the second author by Sofya Raskhodnikova. And while this algorithm and other adaptive monotonicity testers have been considered for quite some time now, to the best of our knowledge Theorem 2 yields the first separation for the query complexity of adaptive and non-adaptive monotonicity testers over any class of Boolean functions over the hypercube. (See for example the discussions in [17, §1.5], [6] and [2].)

Organization.

We discuss the proofs of Theorems 1 and 2 at a high-level in Section 3, after introducing preliminary facts and terminology. The complete proofs follow in Sections 4 and 5, respectively.

2. PRELIMINARIES

2.1 Probability Theory

We use standard concentration inequalities.

LEMMA 3 (Hoeffding's Inequality). *Let $w \in \mathbb{R}^n$ be any real-valued vector. Then for any $t > 0$, when X_1, \dots, X_n are independent random variables taking the values $+1$ and -1 with probability $\frac{1}{2}$ each,*

$$\Pr \left[\left| \sum_{i \in [n]} w_i X_i \right| > t \right] \leq 2e^{-\frac{t^2}{2\|w\|_2^2}},$$

where $\|w\|_2 = \sqrt{\sum_{i \in [n]} w_i^2}$ is the ℓ_2 -norm.

² In fact, we can restrict \mathcal{A} to only query the value of the function on inputs from the middle layers of the hypercube, so it also ϵ -tests *truncated* regular LTFs for monotonicity. See Definition 7 for the definition of truncation, and Section 5.4 for more details.

LEMMA 4 (BERNSTEIN’S INEQUALITY). *Consider a set of n independent random variables X_1, \dots, X_n , where $-1 \leq X_i \leq 1$ for all i . Let $X = \sum_{i \in [n]} X_i$. Then, for all $0 < t < \text{Var}[X]$, we have*

$$\Pr\left[|X - \mathbb{E}[X]| > t\right] \leq 2e^{-\frac{t^2}{4 \text{Var}[X]}}.$$

We also use an anti-concentration inequality that follows directly from the Berry–Esséen theorem. (See, e.g., [21].)

LEMMA 5 (BERRY—ESSÉEN COROLLARY). *Fix $\tau > 0$. Let $w \in \mathbb{R}^n$ be any real-valued vector that satisfies $\max_j |w_j| \leq \tau \|w\|_2$. Then for any $a < b \in \mathbb{R}$, when X_1, \dots, X_n are independent random variables taking the values $+1$ and -1 with probability $\frac{1}{2}$ each,*

$$\Pr\left[a \leq \sum_{i \in [n]} w_i X_i \leq b\right] \leq \frac{b-a}{\|w\|_2} + 2\tau.$$

2.2 Property Testing Lower Bounds

Theorem 1 is established via a standard lemma concerning the general setting where \mathcal{P} and \mathcal{N} are two disjoint families of n -variate Boolean functions, an algorithm is given oracle access to a function $f \in \mathcal{P} \cup \mathcal{N}$, and its task is to determine whether $f \in \mathcal{P}$ or $f \in \mathcal{N}$. The following lemma is essentially folklore—see, e.g., [13] for usage in property testing and [23] for a related lemma. We include a short proof for completeness.

LEMMA 6. *Let Yes and No be probability distributions on n -variate Boolean functions satisfying*

$$\Pr_{f \sim \text{Yes}}[f \in \mathcal{P}] = 1 \quad \text{and} \quad \Pr_{g \sim \text{No}}[g \in \mathcal{N}] = \Omega(1).$$

Assume q is a positive integer such that for any sequences $x_1, \dots, x_q \in \{0, 1\}^n$ and $b_1, \dots, b_q \in \{0, 1\}$,

$$\begin{aligned} & \Pr_{f \sim \text{Yes}}[\forall i: f(x_i) = b_i] \\ & \leq (1 + o(1)) \Pr_{g \sim \text{No}}[\forall i: g(x_i) = b_i] + o(2^{-q}). \end{aligned} \quad (1)$$

Then any randomized algorithm that decides whether $f \in \mathcal{P}$ or $f \in \mathcal{N}$ makes $\Omega(q)$ queries to f .

PROOF. Let \mathcal{A} be a randomized decision tree that distinguishes \mathcal{P} from \mathcal{N} . Denote $p = \Pr_{g \sim \text{No}}[g \in \mathcal{N}] = \Omega(1)$. With a constant number of repetitions of \mathcal{A} , we may assume that \mathcal{A} accepts any function $f \in \mathcal{P}$ with probability at least $1 - p/2$, and accepts each $g \in \mathcal{N}$ with probability at most $1/3$. Then,

$$\Pr_{f \sim \text{Yes}}[\mathcal{A} \text{ accepts on } f] \geq 1 - \frac{p}{2}$$

and

$$\Pr_{g \sim \text{No}}[\mathcal{A} \text{ accepts on } g] \leq (1 - p) + \frac{p}{3} = 1 - \frac{2p}{3},$$

Assume towards a contradiction that \mathcal{A} makes at most q queries. As \mathcal{A} is a probability distribution on deterministic decision trees, there exists a decision tree \mathcal{D} of depth at most q such that

$$\Pr_{f \sim \text{Yes}}[\mathcal{D} \text{ accepts on } f] - \Pr_{g \sim \text{No}}[\mathcal{D} \text{ accepts on } g] \geq \frac{p}{6}. \quad (2)$$

Without loss of generality, we may assume that every leaf of \mathcal{D} is at depth exactly q . Let L denote the set of leaves of \mathcal{D} . Each leaf $\ell \in L$ is characterized by two sequences $x_1, \dots, x_q \in \{0, 1\}^n$ and $b_1, \dots, b_q \in \{0, 1\}$ such that \mathcal{D} ends its work in ℓ on f iff $f(x_i) = b_i$ for all i . Let $L_1 \subseteq L$ be the set of leaves on which \mathcal{D} accepts. Then, by (1),

$$\begin{aligned} \Pr_{f \sim \text{Yes}}[\mathcal{D} \text{ accepts on } f] &= \sum_{\ell \in L_1} \Pr_{f \sim \text{Yes}}[\mathcal{D} \text{ terminates in } \ell \text{ on } f] \\ &\leq (1 + o(1)) \sum_{g \sim \text{No}} \Pr_{\ell \in L_1}[\mathcal{D} \text{ terminates in } \ell \text{ on } g] + o(|L_1|2^{-q}) \\ &= \Pr_{g \sim \text{No}}[\mathcal{D} \text{ accepts on } g] + o(1), \end{aligned}$$

contradicting (2). Hence, \mathcal{A} makes $\Omega(q)$ queries. \square

The following operation is often useful in lower bounds on monotonicity on the hypercube. It essentially reduces monotonicity testing on the whole hypercube to monotonicity testing on its middle layers. This idea comes from [14].

DEFINITION 7. *For $\delta > 0$, the δ -truncation of the function $f: \{0, 1\}^n \rightarrow \{0, 1\}$ is the function $\text{Truncate}_\delta(f)$ defined by*

$$x \mapsto \begin{cases} 0, & \text{if } |x| < \frac{n}{2} - \delta\sqrt{n}; \\ f(x), & \text{if } \frac{n}{2} - \delta\sqrt{n} \leq |x| \leq \frac{n}{2} + \delta\sqrt{n}; \\ 1, & \text{if } |x| > \frac{n}{2} + \delta\sqrt{n}; \end{cases}$$

When f is monotone, then $\text{Truncate}_\delta(f)$ is also monotone. Furthermore, for every $\epsilon > 0$, there exists $\delta > 0$ such that $\text{Truncate}_\delta(f)$ is $\frac{\epsilon}{2}$ -far from monotone whenever f is ϵ -far from monotone. Note that it only makes sense to query $\text{Truncate}_\delta(f)$ on the inputs $x \in \{0, 1\}^n$ satisfying $|x| = \frac{n}{2} \pm O(\sqrt{n})$, since otherwise the response is known in advance. We call such inputs *nearly balanced*.

2.3 Linear Threshold Functions

In studying linear threshold functions, it is more convenient to assume that the function is of the form $f: \{-1, 1\}^n \rightarrow \{-1, 1\}$.

DEFINITION 8. *The function $f: \{-1, 1\}^n \rightarrow \{-1, 1\}$ is a linear threshold function (alternatively: LTF, or halfspace) with associated weights $w_1, \dots, w_n \in \mathbb{R}$ and threshold θ if it satisfies*

$$f(x) = \text{sgn}\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

for every $x \in \{-1, 1\}^n$ where sgn is the sign function defined by $\text{sgn}(x) = 1[x \geq 0]$.

DEFINITION 9 (REGULAR LTF). *LTF $f: \{-1, 1\}^n \rightarrow \{-1, 1\}^n$ is τ -regular if it can be represented with a set of weights w_1, \dots, w_n that satisfy*

$$\max_{i \in [n]} |w_i| \leq \tau \cdot \sqrt{\sum_{i=1}^n w_i^2}.$$

2.4 Noise Sensitivity and Talagrand’s Random DNFs

Let $\mathcal{B}(n, \delta)$ be the probability distribution on the subsets of $[n]$ in which each element is included in the subset independently with probability δ .

DEFINITION 10. The noise sensitivity of a Boolean function $f: \{0,1\}^n \rightarrow \{0,1\}$ at noise rate δ is

$$\text{NS}_\delta(f) = \Pr_{x \sim \{0,1\}^n, S \sim \mathcal{B}(n,\delta)} [f(x) \neq f(x^S)],$$

where x^S denotes the input string x with the variables in S flipped.

Talagrand's random DNF on n variables [24] is a disjunction of $2^{\sqrt{n}}$ independent random conjunctive clauses of size \sqrt{n} .³ More precisely, let \mathcal{C} be the uniform probability distribution on functions $C: [\sqrt{n}] \rightarrow [n]$. We identify each C in \mathcal{C} with the Boolean function $f_C: \{0,1\}^n \rightarrow \{0,1\}$ given by $f_C(x) = \bigwedge_{a \in [\sqrt{n}]} x_{C(a)}$. Talagrand's random DNF f is then defined as

$$f(x) = \bigvee_{j \in [2^{\sqrt{n}}]} f_{C_j}(x),$$

where each clause C_j is independently sampled from \mathcal{C} . Let us denote the distribution of n -variate Talagrand's random DNF by Tal .

One of the particularly useful characteristics of Talagrand's random DNF is that it is one of the most noise-sensitive monotone functions, as shown by the following result.⁴

THEOREM 11 (MOSSEL-O'DONNELL [20]). Talagrand's random DNF f satisfies $\text{NS}_{1/\sqrt{n}}(f) = \Omega(1)$ with probability $\Omega(1)$.

3. HIGH-LEVEL OVERVIEW AND INTUITION

3.1 Bisection Algorithm and Regular LTFs

The intuition behind the proofs of Theorems 1 and 2 is best described by first examining the previous non-adaptive query complexity lower bounds of Chen *et al.* [10, 9]. In these lower bounds, two distributions \mathcal{D}_{Yes} and \mathcal{D}_{No} over a finite set of weights are defined under the two constraints that

1. Every weight in the support of \mathcal{D}_{Yes} is non-negative, and
2. A weight $w \sim \mathcal{D}_{\text{No}}$ is negative with constant probability.

Two distributions **Yes** and **No** over n -variate LTFs are defined by drawing weights w_1, \dots, w_n independently at random from the distributions \mathcal{D}_{Yes} and \mathcal{D}_{No} , respectively, and then by letting

$$f(x_1, \dots, x_n) = \text{sgn}(w_1 x_1 + \dots + w_n x_n).$$

Since \mathcal{D}_{Yes} and \mathcal{D}_{No} are over finite domains (of size independent of n), the resulting function f is always an $O(\frac{1}{\sqrt{n}})$ -regular LTF [9, Claim B.2]. Furthermore, the functions drawn from \mathcal{D}_{Yes} are always monotone, and the functions drawn from \mathcal{D}_{No} are $\Omega(1)$ -far from monotone with large probability [9, Theorem B.9]. Thus, we have the following consequence:

³Talagrand's original definition was for random CNFs. However, DNFs are more convenient than CNFs for our intended applications, and all the results about CNFs easily carry over to the DNF case by duality.

⁴Mossel and O'Donnell only postulate the existence of one such function f . However, Theorem 11 easily follows from the equation before the Proof of Theorem 3 in Section 4 of [20].

THEOREM 12 (CHEN-DE-SERVEDIO-TAN [9]). For all $\delta > 0$, there exist $\epsilon, \tau = \Theta(1)$ such that $\Omega(n^{1/2-\delta})$ non-adaptive nearly balanced queries are required to ϵ -test $\frac{\tau}{\sqrt{n}}$ -regular LTFs for monotonicity.

Regular LTFs are used in the proofs of [10, 9] because with suitable weight distributions \mathcal{D}_{Yes} and \mathcal{D}_{No} , appropriate central limit theorems can be used to bound the query complexity of non-adaptive algorithms. Regular LTFs, however, also have one other notable characteristic: when a $O(\frac{1}{\sqrt{n}})$ -regular LTF is far from monotone, then a *constant* fraction of the edges $x \preceq y$ of the hypercube on which $f(x) \neq f(y)$ are edges where $f(x) > f(y)$ and are thus witnesses to the non-monotonicity of f .

This observation suggests a natural approach for testing monotonicity of regular LTFs: draw an edge $x \preceq y$ uniformly at random from the set of edges where $f(x) \neq f(y)$, and test whether f is monotone on this edge. While we unfortunately do not know of any query-efficient algorithm for drawing edges from this distribution, we do know of one way to at least guarantee that we return *some* edge $x \preceq y$ on which $f(x) \neq f(y)$ using a logarithmic number of queries when f is not too biased. A simple way to do this is described in the bisection algorithm below. In this algorithm, for $x, y \in \{0,1\}^n$, $\text{Hybrid}(x, y)$ denotes the set of inputs $z \in \{0,1\}^n$ that satisfy $z_i = x_i$ for every index $i \in [n]$ where $x_i = y_i$.

Algorithm 1 Bisection algorithm

- 1: Draw $x, y \in \{0,1\}^n$ uniformly and independently at random until $f(x) = 0$ and $f(y) = 1$.
 - 2: If $O(1/\epsilon)$ pairs are drawn without satisfying the condition, **accept**.
 - 3: **while** $|\text{Hybrid}(x, y)| > 2$ **do**
 - 4: Draw $z \in \text{Hybrid}(x, y)$ uniformly at random.
 - 5: If $f(z) = 0$, update $x \leftarrow z$.
 - 6: Otherwise if $f(z) = 1$, update $y \leftarrow z$.
 - 7: **end while**
 - 8: If $x \preceq y$, **accept**; otherwise **reject**.
-

The proof of Theorem 2 is completed by showing that a slight variant of this algorithm does indeed identify a non-monotone edge with constant probability when the input function is a regular LTF that is far from monotone. Specifically, we consider the random process on subsets of $[n]$ defined by the bisection algorithm and show that with constant probability, after $\log n - \Theta(1)$ iterations of the while loop, the set $\{i \in [n] : x_i \neq y_i\}$ has cardinality $O(1)$ and contains some coordinates with negative weights. The details are in Section 5.

3.2 Noise Sensitivity and Polynomial Lower Bound

Theorem 2 shows that we need other functions than regular LTFs to prove a polynomial lower bound for adaptive monotonicity testing. To find such functions, we can start by identifying functions that are far from monotone but for which the bisection algorithm rejects only with small probability.

On a function $f: \{0,1\}^n \rightarrow \{0,1\}$, the bisection algorithm ends its work in an edge xy of the hypercube, where $f(x) \neq f(y)$. Let us say in this case that the algorithm ends its work in variable i , where i is the only variable where x and

y differ. Thus, on each f , the bisection algorithm defines the corresponding *output probability distribution* on the variables in $[n]$. Our first observation is that negating some input variables of a function does not affect the output probability distribution of the bisection algorithm.

PROPOSITION 13. *For each $f: \{0, 1\}^n \rightarrow \{0, 1\}$ and $S \subseteq [n]$, the output probability distributions on $[n]$ defined by the bisection algorithm on the functions f and $g(x) = f(x^S)$ are identical.*

PROOF. Let

$$(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t)$$

be a transcript of the bisection algorithm on the function f . That is, (x_i, y_i) is the value of x and y before the i th iteration of the loop in Algorithm 1. Then,

$$(x_1^S, y_1^S), (x_2^S, y_2^S), \dots, (x_t^S, y_t^S)$$

is an equiprobable transcript of the bisection algorithm on the function g , which ends its work in the same variable. \square

Our next observation is that if we have a monotone function with large noise sensitivity, then negating a (small) random subset of the variables yields a function that is far from monotone with high probability.

LEMMA 14. *Let $f: \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone function and $0 < \delta < 1$ be a real number. Assume $\text{NS}_\delta(f) = \Omega(1)$. Then, with probability $\Omega(1)$ over the choice of $S \sim \mathcal{B}(n, \delta)$, the function $g(x) = f(x^S)$ is $\Omega(1)$ -far from being monotone.*

PROOF. By the definition of noise sensitivity,

$$\Pr_{x \sim \{0, 1\}^n, S \sim \mathcal{B}(n, \delta)} [f(x) \neq f(x^S)] = \Omega(1).$$

By Markov's inequality, with probability $\Omega(1)$ over the choice of $S \sim \mathcal{B}(n, \delta)$, we have

$$\Pr_{x \sim \{0, 1\}^n} [f(x) \neq f(x^S)] = \Omega(1). \quad (3)$$

Let $g(x) = f(x^S)$ be defined for such an S , and let $D(g)$ denote the number of inputs on which we have to modify the value of g in order to make it monotone. We aim to estimate $D(g)$.

Write $x = (y, z)$ with $y \in \{0, 1\}^{[n] \setminus S}$ and $z \in \{0, 1\}^S$. For each y , consider the function $g_y(z) = g(y, z)$. We have $D(g) \geq \sum_y D(g_y)$. Next, each g_y is anti-monotone. This implies $D(g_y) \geq \min\{g_y^{-1}(0), g_y^{-1}(1)\}$. We can lower bound the latter quantity by the number of pairs $\{z, z^S\}$ satisfying $g_y(z) \neq g_y(z^S)$. Summing over all y , we get that $D(g)$ is at least the number of pairs $\{x, x^S\}$ satisfying $f(x) \neq f(x^S)$. By (3), g is $\Omega(1)$ -far from being monotone. \square

These observations, along with Theorem 11, show that there are indeed functions that are far from monotone but are rejected by the bisection algorithm with only a small probability.

PROPOSITION 15. *There exists a function $g: \{0, 1\}^n \rightarrow \{0, 1\}$ that is $\Omega(1)$ -far from being monotone, but such that the bisection algorithm rejects g with probability only $O(1/\sqrt{n})$.*

PROOF. Let f be a monotone Boolean function satisfying $\text{NS}_{1/\sqrt{n}}(f) = \Omega(1)$. By Theorem 11, a Talagrand's random DNF satisfies this condition with probability $\Omega(1)$. Let p_i be the output probability of variable $i \in [n]$ defined by the bisection algorithm on f . Let $S \sim \mathcal{B}(n, 1/\sqrt{n})$. Then,

$$\mathbb{E}_S \left[\sum_{i \in S} p_i \right] = \frac{1}{\sqrt{n}}.$$

By Markov's inequality, and using Lemma 14, there exists S such that the function $g(x) = f(x^S)$ is $\Omega(1)$ -far from being monotone, and $\sum_{i \in S} p_i = O(1/\sqrt{n})$. By Proposition 13, the latter sum is exactly equal to the rejection probability of the bisection algorithm on the function g . \square

This result shows that there are functions obtained by negating some variables in a Talagrand's random DNF that are $\Omega(1)$ -far from monotone, but such that the bisection algorithm requires $\Omega(\sqrt{n})$ queries to detect that it is non-monotone. The proof of Theorem 1 uses a very different approach—after all, there is no direct analogue of Proposition 13 that can hold for all adaptive algorithms—but the underlying ideas are the same. We show that for *any* set of $q \ll n^{1/4}$ queries, the distribution of the values returned by the monotone and the non-monotone Talagrand's random DNFs are very similar. After that, we can apply Lemma 6 to complete the proof. The high-level intuition is as follows. Consider two queries $x, y \in \{0, 1\}^n$. On the one hand, if x and y are far from each other, then, due to noise sensitivity, the values of $f(x)$ and $f(y)$ are essentially independent. Hence, adaptivity does not help here. On the other hand, if x and y are close, they are likely to miss the set S of negated input variables. More precisely, since there are at most $q^2 \ll \sqrt{n}$ pairs of close inputs, a random set S of \sqrt{n} elements will avoid all of them with high probability. For all the details, see Section 4.

4. POLYNOMIAL LOWER BOUND

In this section, we prove Theorem 1. Throughout this section we use $\mathcal{B} = \mathcal{B}(n, 1/\sqrt{n})$ to denote the probability distribution on subsets of $[n]$ where each element is included in the subset independently with probability $1/\sqrt{n}$. Following the discussion in Section 3.2, let us define the distribution Tal^\pm of *Talagrand's random non-monotone DNFs* as the following distribution on n -variate Boolean functions

$$\text{Tal}^\pm = \left\{ x \mapsto f(x^S) \mid f \sim \text{Tal}, S \sim \mathcal{B} \right\}.$$

We define two distributions for a sufficiently large constant $\delta > 0$:

$$\text{Yes} = \{ \text{Truncate}_\delta(f) \mid f \sim \text{Tal} \}$$

and

$$\text{No} = \{ \text{Truncate}_\delta(f) \mid f \sim \text{Tal}^\pm \}.$$

In view of Lemma 6, Theorem 11 and Lemma 14, it suffices to show that for all $q = O(n^{1/4} \log^{-2} n)$, nearly balanced input strings $x_1, \dots, x_q \in \{0, 1\}^n$ and Boolean outcomes $b_1, \dots, b_q \in \{0, 1\}$, we have

$$\begin{aligned} & \Pr_{f \sim \text{Tal}} [\forall i: f(x_i) = b_i] \\ & \leq (1 + o(1)) \Pr_{g \sim \text{Tal}^\pm} [\forall i: g(x_i) = b_i] + o(2^{-q}). \end{aligned} \quad (4)$$

4.1 Proof of Theorem 1

Let us denote by $X = \{x_1, \dots, x_q\}$ the set of input strings. All of them are nearly balanced. In this section, we often identify strings in $\{0, 1\}^n$ with the corresponding subsets of $[n]$.

For a fixed sequence of values b_1, \dots, b_q , the set X can be naturally divided into

$$X_0 = \{x_i \mid i \in [q], b_i = 0\} \text{ and } X_1 = \{x_i \mid i \in [q], b_i = 1\}.$$

Let $J = [2^{\sqrt{n}}]$ be the set of indices of the clauses in Talagrand's random DNF. Recall that a function f from the Tal distribution is given by

$$f(x) = \bigvee_{j \in J} f_{C_j}(x),$$

where $(C_j) \sim \mathcal{C}^J$ is a sequence of random clauses. We call the sequence (C_j) *compliant with respect to the shift* $T \subseteq [n]$ iff the corresponding function f satisfies $\forall i \in [q]: f(x_i^T) = b_i$. We denote the set of such sequences by \mathcal{M}^T . The set \mathcal{M}^\emptyset corresponds to the events on the left-hand side of (4), and \mathcal{M}^S for $S \sim \mathcal{B}$ corresponds to the right-hand side.

For $T \subseteq [n]$, we partition the set \mathcal{M}^T in accordance to when a clause in the sequence (C_j) first satisfies each particular input $x \in X_1$. Formally, for each sequence $\tau = (\tau_x) \in J^{X_1}$, we define \mathcal{M}_τ^T as the set of sequences $(C_j) \in \mathcal{M}^T$ satisfying

- for each $x \in X_0$, we have $f_{C_j}(x^T) = 0$ for all $j \in J$;
- for each $x \in X_1$, we have $f_{C_j}(x^T) = 0$ for all $j < \tau_x$, and $f_{C_{\tau_x}}(x^T) = 1$.

This clearly partitions the set \mathcal{M}^T into disjoint subsets.

The conditions imposed by (5) on different C_j are independent, thus, we can decompose \mathcal{M}_τ^T into the following Cartesian product:

$$\mathcal{M}_\tau^T = \prod_{j \in J} \mathcal{M}_{\tau,j}^T, \quad (6)$$

where $\mathcal{M}_{\tau,j}^T$ is the projection of \mathcal{M}_τ^T onto the j th component of the sequence. Let, for $j \in J$,

$$X_{1,j} = \{x \in X_1 \mid \tau_x = j\}$$

and

$$X_{0,j} = X_0 \cup \{x \in X_1 \mid j < \tau_x\}.$$

Thus, for each j , we have $C \in \mathcal{M}_{\tau,j}^T$ if and only if $f_C(x^T) = 1$ for all $x \in X_{1,j}$ and $f_C(x^T) = 0$ for all $x \in X_{0,j}$.

We say that a sequence τ is *good* if

$$\forall j \in J, \forall x, y \in X_{1,j}: |x \cap y| \geq \frac{n}{2} - n^{3/4}, \quad (7)$$

Otherwise, we call it *bad*. We treat these two cases separately.

LEMMA 16. *We have*

$$\Pr_{(C_j) \sim \mathcal{C}^J} [\text{exists bad } \tau \text{ such that } (C_j) \in \mathcal{M}_\tau^\emptyset] = o(2^{-q}). \quad (8)$$

PROOF. It is easy to see that any (C_j) satisfying the condition in (8) also satisfies

$$\begin{aligned} \exists j \in J, \exists x, y \in X: \\ \left(|x \cap y| < \frac{n}{2} - n^{3/4} \right) \wedge \left(f_{C_j}(x) = f_{C_j}(y) = 1 \right). \end{aligned} \quad (9)$$

By the union bound, the probability that (C_j) satisfies (9) is at most

$$\begin{aligned} 2^{\sqrt{n}} q^2 \left(\frac{\frac{n}{2} - n^{3/4}}{n} \right)^{\sqrt{n}} &= q^2 (1 - 2n^{-1/4})^{\sqrt{n}} \leq q^2 e^{-2n^{1/4}} \\ &= q^2 e^{-\Omega(\log^2 n)} \cdot 2^{-q} = o(2^{-q}). \quad \square \end{aligned}$$

Let us now consider good τ . In order to prove (4), it suffices to show that

$$\prod_{j \in J} |\mathcal{M}_{\tau,j}^\emptyset| \leq (1 + o(1)) \prod_{j \in J} |\mathcal{M}_{\tau,j}^S| \quad (10)$$

with probability $1 - o(1)$ over the choice of $S \sim \mathcal{B}$. Indeed, using (6), we get from (10) that

$$\begin{aligned} \sum_{\tau \text{ is good}} |\mathcal{M}_\tau^\emptyset| &\leq (1 + o(1)) \sum_{\tau \text{ is good}} \mathbb{E}_{S \sim \mathcal{B}} [|\mathcal{M}_\tau^S|] \\ &\leq (1 + o(1)) \mathbb{E}_{S \sim \mathcal{B}} [|\mathcal{M}^S|] \\ &= (1 + o(1)) \cdot |\mathcal{C}^J| \Pr_{g \sim \text{Tal}^\pm} [\forall i: g(x_i) = b_i]. \end{aligned}$$

Hence, using also Lemma 16, we get

$$\begin{aligned} \Pr_{f \sim \text{Tal}} [\forall i: f(x_i) = b_i] &= \frac{1}{|\mathcal{C}^J|} \left(\sum_{\tau \text{ is good}} |\mathcal{M}_\tau^\emptyset| + \sum_{\tau \text{ is bad}} |\mathcal{M}_\tau^\emptyset| \right) \\ &\leq (1 + o(1)) \Pr_{g \sim \text{Tal}^\pm} [\forall i: g(x_i) = b_i] + o(2^{-q}). \end{aligned}$$

Let us consider (10) now. The set of indices J breaks down into two parts $J = J_1 \cup J_0$, where

$$J_1 = \{\tau_x \mid x \in X_1\} \quad \text{and} \quad J_0 = J \setminus J_1.$$

We prove (10) for the indices in J_1 and J_0 independently. Indices in J_0 are easier to analyse because for them we have $X_{1,j} = \emptyset$. On the other hand, we need a rather careful estimate since $|J_0| \approx 2^{\sqrt{n}}$. Indices in J_1 are harder to analyse because for them, in general, both $X_{1,j}$ and $X_{0,j}$ are non-empty. But since $|J_1| \leq q < n^{1/4}$, a less accurate estimate suffices.

For J_1 , we have the following lemma, which is proven in Section 4.3.

LEMMA 17. *Assume τ is good. Then, for a set $S \sim \mathcal{B}$, with probability $1 - o(1)$, we have*

$$\prod_{j \in J_1} |\mathcal{M}_{\tau,j}^\emptyset| \leq (1 + o(1)) \prod_{j \in J_1} |\mathcal{M}_{\tau,j}^S|. \quad (11)$$

For J_0 , we have the following lemma, which we prove in Section 4.4:

LEMMA 18. *A set $S \sim \mathcal{B}$ satisfies the following property with probability $1 - o(1)$: For any subset $X' \subseteq X$, we have*

$$\begin{aligned} \Pr_{\mathcal{C} \sim \mathcal{C}} [\exists x \in X': f_C(x) = 1] \\ \geq \left(1 - O(n^{-1/4} \log^{3/2} n) \right) \Pr_{\mathcal{C} \sim \mathcal{C}} [\exists x \in X': f_C(x^S) = 1]. \end{aligned} \quad (12)$$

Assume S satisfies (11) and (12). For $j \in J_0$, let p_j and p'_j denote the probability in the left- and right-hand sides

of (12), respectively, when $X' = X_{0,j}$. By the union bound, and since all $x \in X$ are nearly balanced,

$$\begin{aligned} p'_j &\leq \sum_{x \in X_{0,j}} \Pr[f_C(x^S) = 1] \\ &\leq |X_{0,j}| \left(\frac{\frac{n}{2} + O(\sqrt{n})}{n} \right)^{\sqrt{n}} = O(q2^{-\sqrt{n}}). \end{aligned}$$

Thus,

$$\begin{aligned} \frac{|\mathcal{M}_{\tau,j}^0|}{|\mathcal{M}_{\tau,j}^S|} &= \frac{1 - p_j}{1 - p'_j} \leq \frac{1 - (1 - O(n^{-1/4} \log^{3/2} n))p'_j}{1 - p'_j} \\ &= 1 + O(n^{-1/4} \log^{3/2} n \cdot q2^{-\sqrt{n}}) \\ &= 1 + o(2^{-\sqrt{n}}). \end{aligned} \quad (13)$$

Hence, taking the product of (13) over all $j \in J_0$, we get that, with probability $1 - o(1)$, a set $S \sim \mathcal{B}$ satisfies:

$$\prod_{j \in J_0} \frac{|\mathcal{M}_{\tau,j}^0|}{|\mathcal{M}_{\tau,j}^S|} \leq (1 + o(2^{-\sqrt{n}}))^{2\sqrt{n}} = 1 + o(1).$$

Multiplying this by (11), we obtain (10).

4.2 A Simple Lemma

In this section, we prove a simple lemma that will be used in the proofs of both Lemmata 17 and 18. Let $\gamma = \omega(\sqrt{n})$. Define a graph G on the vertex set X defined in Section 4.1, where two vertices x and y are connected iff $|x \cap y| \geq n/2 - \gamma$.

LEMMA 19. *For every non-empty connected subset A of vertices of G , we have*

$$\left| \bigcap_{x \in A} x \right| \geq \frac{n}{2} - O(|A|\gamma) \quad \text{and} \quad \left| \bigcup_{x \in A} x \right| \leq \frac{n}{2} + O(|A|\gamma). \quad (14)$$

PROOF. We prove the first equality in (14), the second one being similar. The proof is by induction on the size of A . The base case $|A| = 1$ follows from the fact that $x \in X$ is nearly balanced.

For the inductive step, take a vertex $y \in A$ such that $A \setminus \{y\}$ is connected. Let z be a neighbour of y in $A \setminus \{y\}$. By the inductive hypothesis,

$$\left| \bigcap_{x \in A \setminus \{y\}} x \right| \geq \frac{n}{2} - O((|A| - 1)\gamma).$$

Also,

$$\left| \bigcap_{x \in A \setminus \{y\}} x \right| - \left| \bigcap_{x \in A} x \right| \leq |z| - |z \cap y| = O(\gamma),$$

since z is nearly balanced and $|y \cap z| \geq n/2 - \gamma$. Combining the last two inequalities, we obtain (14). \square

4.3 Proof of Lemma 17

As we only work with J_1 in this section, let, for $T \subseteq [n]$,

$$\mathcal{M}_1^T = \prod_{j \in J_1} \mathcal{M}_j^T.$$

This is the projection of \mathcal{M}^T from Section 4.1 onto the indices in J_1 . For $j \in J_1$, let us denote

$$y_j = \bigcap_{x \in X_{j,1}} x, \quad \text{and} \quad z_j = \bigcup_{x \in X_{j,1}} x.$$

Using Lemma 19 with $\gamma = n^{3/4}$, we get

$$\frac{n}{3} \leq \frac{n}{2} - O(qn^{3/4}) \leq |y_j| \leq \frac{n}{2} + O(n^{3/4})$$

and

$$|z_j| \leq \frac{n}{2} + O(|X_{1,j}|n^{3/4}),$$

if n is large enough. We impose the following constraints on $S \sim \mathcal{B}$:

$$\begin{aligned} |S| &= O(\sqrt{n}), \\ |S \cap y_j| &\leq \frac{\sqrt{n}}{2} + O(n^{1/4} \sqrt{\log n}) \\ |S \setminus z_j| &\geq \frac{\sqrt{n}}{2} - O(|X_{1,j}|n^{1/4} \sqrt{\log n}) \end{aligned} \quad (15)$$

for all $j \in J_1$. For $S \setminus z_j$, we have

$$\mathbb{E}[|S \setminus z_j|] \geq \frac{\sqrt{n}}{2} - O(|X_{1,j}|n^{1/4}) \quad \text{and} \quad \text{Var}[|S \setminus z_j|] \leq \sqrt{n}.$$

And similar estimates can be obtained for $S \cap y_j$. Applying Bernstein's inequality and the union bound, we get that a set $S \sim \mathcal{B}$ satisfies (15) with probability $1 - o(1)$ if the O factors in (15) are large enough. In the remaining part of this section, we assume that S satisfies (15).

For each $(C_j) \in \mathcal{M}_1^0$, we have $C_j(a) \in y_j$ for each $j \in J_1$ and $a \in [\sqrt{n}]$. Also, for each $j \in J_1$ and $x \in X_{0,j}$, there exists $a \in [\sqrt{n}]$ such that $C_j(a) \notin x$. We call the smallest such a the *pivotal index* corresponding to j and x . We call $C_j(a)$ the corresponding *pivotal element*.

Let S_j be an arbitrary subset of $S \cap y_j$ of size

$$|S_j| = \max\{0, |S \cap y_j| - |S \setminus z_j|\} = O(|X_{1,j}|n^{1/4} \sqrt{\log n}). \quad (16)$$

We define two auxiliary subsets of \mathcal{C}^{J_1} .

- A sequence $(C_j) \in \mathcal{M}_1^0$ is called *half-restricted* if all its pivotal elements lie outside of S . Denote the set of half-restricted (C_j) by \mathcal{H}^S .
- A sequence $(C_j) \in \mathcal{H}^S$ is called *restricted* if for all $j \in J_1$ and $a \in [\sqrt{n}]$ we have $C_j(a) \notin S_j$. Denote the set of restricted (C_j) by \mathcal{R}^S .

Lemma 17 follows from the following three claims.

CLAIM 20. *We have $|\mathcal{M}_1^S| \geq |\mathcal{R}^S|$.*

PROOF. This is achieved by *shifting*: moving elements from $S \cap y_j$ to $S \setminus z_j$. More precisely, let $\pi_j: S \cap y_j \setminus S_j \rightarrow S \setminus z_j$ be any injective mapping. It exists due to (16). Define a mapping $\pi: \mathcal{R}^S \rightarrow \mathcal{M}_1^S$ as $\pi: (C_j) \mapsto (C'_j)$, where

$$C'_j(a) = \begin{cases} \pi_j(C_j(a)), & \text{if } C_j(a) \in S \cap y_j \setminus S_j; \\ C_j(a), & \text{if } C_j(a) \in y_j \setminus S. \end{cases}$$

It is clearly an injective mapping. Also, its image is a subset of \mathcal{M}_1^S since,

- $C'_j(a) \in x^S$ for each $j \in J_1$, $a \in [\sqrt{n}]$ and $x \in X_{1,j}$; and
- for each pivotal element $C_j(a)$ corresponding to $j \in J_1$ and $x \in X_{0,j}$, we have $C'_j(a) = C_j(a) \notin x^S$, ensuring $f_{C'_j}(x^S) = 0$. \square

CLAIM 21. *With probability $1 - o(1)$ over the choice of $S \sim \mathcal{B}$, we have $|\mathcal{M}_1^0| \leq (1 + o(1))|\mathcal{H}^S|$.*

PROOF. For each $i \in [n]$, let d_i denote the number of sequences $(C_j) \in \mathcal{M}_1^\emptyset$ for which i is a pivotal element. Since each (C_j) has at most q^2 pivotal elements, we see that

$$\sum_{i \in [n]} d_i \leq |\mathcal{M}_1^\emptyset| q^2 = o(|\mathcal{M}_1^\emptyset| \sqrt{n}).$$

In particular, $\mathbb{E}_{S \sim \mathcal{B}} \left[\sum_{i \in S} d_i \right] = o(|\mathcal{M}_1^\emptyset|)$. By Markov's inequality, with probability $1 - o(1)$, we have $\sum_{i \in S} d_i = o(|\mathcal{M}_1^\emptyset|)$, implying the claim. \square

CLAIM 22. We have $|\mathcal{H}^S| \leq (1 + o(1))|\mathcal{R}^S|$.

PROOF. In this case, it is easier to consider each $j \in J_1$ independently. Again, the conditions for different j are independent, hence,

$$|\mathcal{R}^S| = \prod_{j \in J_1} |\mathcal{R}_j^S|, \quad \text{and} \quad |\mathcal{H}^S| = \prod_{j \in J_1} |\mathcal{H}_j^S|,$$

where \mathcal{R}_j^S and \mathcal{H}_j^S are the projections of \mathcal{R}^S and \mathcal{H}^S onto the j th component. We prove that

$$|\mathcal{H}_j^S| \leq e^{O(n^{-1/4} \sqrt{\log n} |X_{1,j}|)} |\mathcal{R}_j^S|, \quad (17)$$

which implies the claim, since then

$$\begin{aligned} \frac{|\mathcal{H}^S|}{|\mathcal{R}^S|} &\leq e^{O(n^{-1/4} \sqrt{\log n} \sum_{j \in J_1} |X_{1,j}|)} \\ &\leq e^{O(n^{-1/4} \sqrt{\log n} \cdot q)} = 1 + o(1). \end{aligned}$$

Let $\mathcal{H}_{j,k}^S$ denote the subset of $C' \in \mathcal{H}_j^S$ such that for exactly k values of $a \in [\sqrt{n}]$, we have $C'(a) \in S_j$. In particular, $\mathcal{R}_j^S = \mathcal{H}_{j,0}^S$, and $\mathcal{H}_j^S = \bigcup_k \mathcal{H}_{j,k}^S$. We say that a clause $C \in \mathcal{R}_j^S$ is *in relation* with a clause $C' \in \mathcal{H}_{j,k}^S$ iff $C(a) = C'(a)$ whenever $C'(a) \notin S_j$.

Clearly, each clause $C \in \mathcal{R}_j^S$ is in relation with at most $\binom{\sqrt{n}}{k} |S_j|^k$ clauses in $\mathcal{H}_{j,k}^S$. Next, we claim that if we take a clause $C' \in \mathcal{H}_{j,k}^S$ and substitute each $C'(a) \in S_j$ with an element of $y_k \setminus S$, we get a clause $C \in \mathcal{R}_j^S$. First, any $C'(a)$ that was changed was not a pivotal element of C' since $S_j \subseteq S$ and $C' \in \mathcal{H}_j^S$. Next, a new element $C(a)$ can become a pivotal element of C , but it lies outside of S , so, nonetheless, $C \in \mathcal{R}_j^S$. Hence, each clause $C' \in \mathcal{H}_{j,k}^S$ is in relation with at least $|y_j \setminus S|^k$ clauses in \mathcal{R}_j^S . Using double counting,

$$\begin{aligned} \frac{|\mathcal{H}_{j,k}^S|}{|\mathcal{R}_j^S|} &\leq \frac{\binom{\sqrt{n}}{k} |S_j|^k}{|y_j \setminus S|^k} \leq \frac{n^{k/2}/k! \cdot (O(|X_{1,j}| n^{1/4} \sqrt{\log n}))^k}{(\Omega(n))^k} \\ &= \frac{1}{k!} \left(O\left(\frac{|X_{1,j}| \sqrt{\log n}}{n^{1/4}}\right) \right)^k. \end{aligned}$$

Hence,

$$\begin{aligned} \frac{|\mathcal{H}_j^S|}{|\mathcal{R}_j^S|} &\leq 1 + \sum_{k \geq 1} \frac{1}{k!} \left(O\left(\frac{|X_{1,j}| \sqrt{\log n}}{n^{1/4}}\right) \right)^k \\ &= e^{O(n^{-1/4} \sqrt{\log n} |X_{1,j}|)}. \quad \square \end{aligned}$$

4.4 Proof of Lemma 18

As in Section 4.1, we treat pairs of inputs that are far from each other separately. Let a parameter $\gamma = \Theta(\sqrt{n} \log n)$ be

specified later. Define the graph G as in Section 4.2. Let G_1, \dots, G_\varkappa be the connected components of G , and

$$z_k = \bigcap_{x \in G_k} x.$$

Using Lemma 19, we get that

$$|z_k| \geq \frac{n}{2} - O(q\gamma) \geq \frac{n}{2} - O(n^{3/4}).$$

We impose the following constraints on $S \sim \mathcal{B}$:

$$\begin{aligned} |S| &= O(\sqrt{n}), \\ |S \cap x| &\geq \frac{\sqrt{n}}{2} - O(n^{1/4} \sqrt{\log n}), \\ |S \setminus z_k| &\leq \frac{\sqrt{n}}{2} + O(n^{1/4} \sqrt{\log n}) \end{aligned} \quad (18)$$

for all $x \in X$ and $k \in [\varkappa]$. By Bernstein's inequality again, if the O factors are chosen appropriately, a set $S \sim \mathcal{B}$ satisfies (18) with probability $1 - o(1)$.

Let us assume up to the end of the section that a subset S satisfying (18) is fixed. We say a clause C is *positive with respect to the shift* $T \in \{\emptyset, S\}$ iff $f_C(x^T) = 1$ for some $x \in X'$. Denote the set of such clauses by \mathcal{P}^T .

We are also going to treat a clause $C: [\sqrt{n}] \rightarrow [n]$ as a multiset. For example, if $A \subseteq [n]$, we denote by $C \cap A$ the partial function from $[\sqrt{n}]$ to $[n]$, defined by $(C \cap A)(a) = C(a)$ for all $a \in [\sqrt{n}]$ such that $C(a) \in A$, and not defined for the remaining a . We call such functions *partial clauses*. A partial clause $C \setminus A$ is defined similarly. The *size* of a partial clause is the size of its domain. We say that a partial clause is contained in $A \subseteq [n]$ if its range is contained in A .

CLAIM 23. We have

$$\Pr_{C \sim \mathcal{P}^S} \left[|C \cap S| \geq \Omega(\log n) \right] \leq \frac{1}{n}.$$

PROOF. This holds because $|C \cap S|$ approximately follows a Poisson distribution. Indeed, for a non-negative integer k , let

$$\mathcal{P}_k^S = \{C \in \mathcal{P}^S \mid |C \cap S| = k\}.$$

We say that $C \in \mathcal{P}_0^S$ is *in relation* with $C' \in \mathcal{P}_k^S$ iff $C(a) = C'(a)$ for all a such that $C'(a) \notin S$.

Each $C \in \mathcal{P}_0^S$ is in relation with at most $\binom{\sqrt{n}}{k} |S|^k$ clauses in \mathcal{P}_k^S . On the other hand, let $C' \in \mathcal{P}_k^S$. Then, there exists $x \in X'$ such that $C' \subseteq x^S$. Hence, C' is in relation with at least $|x \setminus S|^k = (\Omega(n))^k$ clauses in \mathcal{P}_0^S . Using double counting,

$$\frac{|\mathcal{P}_k^S|}{|\mathcal{P}_0^S|} \leq \frac{\binom{\sqrt{n}}{k} (O(\sqrt{n}))^k}{(\Omega(n))^k} = \frac{(O(1))^k}{k!}.$$

This implies the claim. \square

Thus, we can only focus on those C that have small intersection with S . Let B be a partial clause with $B \subseteq [n] \setminus S$ and $\sqrt{n} - O(\log n) \leq |B| \leq \sqrt{n}$. For $T \in \{\emptyset, S\}$, let us denote

$$\mathcal{P}_B^T = \{C \in \mathcal{P}^T \mid C \setminus S = B\}.$$

We say B is *bad* if $B \subseteq x \cap y$ where $x, y \in X'$ are vertices from different connected components of G . Otherwise, we call B *good*.

LEMMA 24. If B is good, then

$$|\mathcal{P}_B^\emptyset| \geq \left(1 - O(n^{-1/4} \log^{3/2} n)\right) |\mathcal{P}_B^S|.$$

PROOF. Let $X_B = \{x \in X' \mid B \subseteq x\}$. If X_B is empty, then both \mathcal{P}_B^\emptyset and \mathcal{P}_B^S are empty, and we are done, so assume there is some $x \in X_B$. Let $y_B = \bigcap_{y \in X_B} y$. Also, as B is good, X_B is contained in some connected component G_k of G . In particular, $z_k \subseteq y_B$.

Let $\bar{D} \subseteq [\sqrt{n}]$ be the complement of the domain of B . In particular, $|\bar{D}| = O(\log n)$. The size of \mathcal{P}_B^\emptyset is at least the number of functions from \bar{D} to $x \cap S$, and the size of \mathcal{P}_B^S is at most the number of functions from \bar{D} to $S \setminus y_B \subseteq S \setminus z_k$. Thus, using (18):

$$\begin{aligned} \frac{|\mathcal{P}_B^\emptyset|}{|\mathcal{P}_B^S|} &\geq \left(\frac{\frac{\sqrt{n}}{2} - O(n^{1/4} \sqrt{\log n})}{\frac{\sqrt{n}}{2} + O(n^{1/4} \sqrt{\log n})} \right)^{O(\log n)} \\ &\geq 1 - O(n^{-1/4} \log^{3/2} n). \quad \square \end{aligned}$$

LEMMA 25. We have

$$\left| \bigcup_{B \text{ is bad}} \mathcal{P}_B^S \right| \leq O\left(\frac{1}{n}\right) |\mathcal{P}^S|.$$

PROOF. Fix a particular pair $x, y \in X'$ of vertices that lie in different connected components of G . Then, for each \mathcal{P}_B^S with $B \subseteq x \cap y$, and each clause $C \in \mathcal{P}_B^S$, we have $C \subseteq (x \cap y) \cup S$. On the other hand, we may lower bound the number of clauses contained in \mathcal{P}^S by the number of clauses contained in x^S . Thus,

$$\begin{aligned} \frac{\left| \bigcup_{B \subseteq x \cap y} \mathcal{P}_B^S \right|}{|\mathcal{P}^S|} &\leq \left(\frac{|(x \cap y) \cup S|}{|x^S|} \right)^{\sqrt{n}} \leq \left(\frac{\frac{n}{2} - \Omega(\gamma)}{\frac{n}{2} - O(\sqrt{n})} \right)^{\sqrt{n}} \\ &\leq \left(1 - \Omega\left(\frac{\gamma}{n}\right) \right)^{\sqrt{n}} \leq e^{-\Omega(\gamma/\sqrt{n})}. \end{aligned} \tag{19}$$

Taking the Θ -factor in the definition of γ sufficiently large and summing (19) over all x and y , we obtain the lemma. \square

Thus, using Lemmata 24, 25 and Claim 23,

$$\begin{aligned} |\mathcal{P}^\emptyset| &\geq \sum_{B \text{ is good}} |\mathcal{P}_B^\emptyset| \geq (1 - O(n^{-1/4} \log^{3/2} n)) \sum_{B \text{ is good}} |\mathcal{P}_B^S| \\ &\geq (1 - O(n^{-1/4} \log^{3/2} n)) |\mathcal{P}^S|, \end{aligned}$$

proving Lemma 18.

5. TESTING MONOTONICITY OF REGULAR LTFS

5.1 Randomized Bisection Process

The key component of the analysis of the bisection algorithm and the proof of Theorem 2 is the analysis of *randomized bisection processes*, as defined below.

DEFINITION 26 (RANDOMIZED BISECTION PROCESS).

Fix any finite set S . The randomized bisection process with initial set S is the sequence of random sets S_0, S_1, S_2, \dots defined as follows. Initially, $S_0 = S$. For each $k \geq 1$, S_{k-1} is partitioned uniformly at random into two sets A_k and B_k . Then the set S_k is chosen to be either A_k or B_k by some arbitrary (and possibly adversarial) external process.

LEMMA 27. For any $\delta > 0$, there exists $\kappa = \kappa(\delta)$ such that with probability at least $1 - \delta$, the randomized bisection process S_0, S_1, S_2, \dots with initial set S satisfies

$$\frac{1}{2} \cdot \frac{|S|}{2^k} < |S_k| < \frac{3}{2} \cdot \frac{|S|}{2^k} \tag{20}$$

for every $k \leq \log |S| - \kappa$.

PROOF. Let us prove the lower bound first. It is clear that the best strategy for the adversary is to take the smallest of A_k and B_k on each step, so we may assume that the sets S_k of size less than $|S_{k-1}|/2$ have double probability to appear, whereas the sets S_k of size more than $|S_{k-1}|/2$ never appear at all.

Using Fubini's theorem and the Chernoff-Hoeffding bound, we obtain, for a fixed S_{k-1} ,

$$\begin{aligned} \mathbb{E}[|S_k|] &= \frac{|S_{k-1}|}{2} - 2 \int_0^{+\infty} \Pr\left[\mathcal{B} < \frac{|S_{k-1}|}{2} - t\right] dt \\ &\geq \frac{|S_{k-1}|}{2} - 2 \int_0^{+\infty} e^{-2t^2/|S_{k-1}|} dt \\ &\geq \frac{|S_{k-1}|}{2} - O\left(\sqrt{|S_{k-1}|}\right), \end{aligned}$$

where \mathcal{B} is the binomial probability distribution on $|S_{k-1}|$ elements with probability $\frac{1}{2}$.

Since $x \mapsto \frac{x}{2} - c\sqrt{x}$ is a convex function, if we unfix S_{k-1} , we get by Jensen's inequality

$$\mathbb{E}[|S_k|] \geq \frac{\mathbb{E}[|S_{k-1}|]}{2} - O\left(\sqrt{\mathbb{E}[|S_{k-1}|]}\right). \tag{21}$$

It is clear that $|S_k| \leq |S|/2^k$, thus, by induction on k ,

$$\begin{aligned} \mathbb{E}[|S_k|] &\geq \frac{|S|}{2^k} - \sum_{j=1}^k \frac{1}{2^{k-j}} \cdot O\left(\sqrt{\frac{|S|}{2^{j-1}}}\right) \\ &= \frac{|S|}{2^k} - \sqrt{\frac{|S|}{2^k}} \sum_{j=1}^k \frac{O(1)}{2^{(k-j)/2}} \\ &\geq \frac{|S|}{2^k} - O\left(\sqrt{\frac{|S|}{2^k}}\right). \end{aligned} \tag{22}$$

If $\frac{|S|}{2^k} = \Omega_\delta(1)$, we have

$$\mathbb{E}[|S_k|] > \frac{|S|}{2^k} - \frac{\delta}{4} \cdot \frac{|S|}{2^k}.$$

And since $|S_k| \leq |S|/2^k$, we have by Markov's inequality that

$$\Pr\left[|S_k| \leq \frac{1}{2} \cdot \frac{|S|}{2^k}\right] \leq \frac{\delta}{2}.$$

The proof of the upper bound is similar. This time the adversary takes the largest of A_k and B_k . Similarly to (21), we get

$$\mathbb{E}[|S_k|] \leq \frac{\mathbb{E}[|S_{k-1}|]}{2} + O\left(\sqrt{\mathbb{E}[|S_{k-1}|]}\right).$$

We show by induction on k that if $\frac{|S|}{2^k} = \Omega(1)$, then $\mathbb{E}[|S_k|] \leq \frac{3}{2} \cdot \frac{|S|}{2^k}$. This is done similarly to (22):

$$\begin{aligned} \mathbb{E}[|S_k|] &\leq \frac{|S|}{2^k} + \sum_{j=1}^k \frac{1}{2^{k-j}} \cdot O\left(\sqrt{\frac{3|S|}{2^j}}\right) \\ &= \frac{|S|}{2^k} + \sqrt{\frac{|S|}{2^k}} \sum_{j=1}^k \frac{O(1)}{2^{(k-j)/2}} \\ &\leq \frac{|S|}{2^k} + O\left(\sqrt{\frac{|S|}{2^k}}\right). \end{aligned}$$

Again, if $\frac{|S|}{2^k} = \Omega_\delta(1)$, we have $\mathbb{E}[|S_k|] < \frac{|S|}{2^k} + \frac{\delta}{4} \cdot \frac{|S|}{2^k}$, and, since $|S_k| \geq |S|/2^k$,

$$\Pr\left[|S_k| \geq \frac{3}{2} \cdot \frac{|S|}{2^k}\right] \leq \frac{\delta}{2}. \quad \square$$

5.2 Non-Monotonicity of LTFs

PROPOSITION 28. *If $f: \{-1, 1\}^n \rightarrow \{-1, 1\}$ is a non-constant LTF with weights w_1, \dots, w_n such that*

$$\sum_{i: w_i < 0} |w_i| > \max_i w_i,$$

then f is not monotone.

PROOF. Let $N = \{i \in [n] \mid w_i < 0\}$ denote the set of indices with negative weights and let $\eta = \sum_{i \in N} |w_i|$. Let $X \in \{-1, 1\}^n$ be the subset of inputs such that for every $i \in N$, $x_i = 1$.

There exists $x \in X$ such that $\theta - 2\eta \leq \sum_{i \in [n]} w_i x_i < \theta$. Indeed, there exists an input $x' \in X$ with $\sum_{i \in [n]} w_i x'_i < \theta$ (otherwise f is the constant 1 function), and an input $x'' \in X$ with $\sum_{i \in [n]} w_i x''_i \geq \theta - 2\eta$ (otherwise f is the constant -1 function). Also, $\max_i |w_i| \leq \eta$, hence, changing the value of one variable changes the value of the sum $\sum_{i \in [n]} w_i x_i$ by at most 2η .

With this choice of x , let $y \in \{-1, 1\}^n$ be defined by $y_i = x_i$ for $i \in [n] \setminus N$ and $y_j = -1$ for every $j \in N$. Then $\sum_{i \in [n]} w_i y_i \geq \theta$ so $x \succeq y$ and $1 = f(y) > f(x) = -1$, hence, f is non-monotone. \square

In the proof of Theorem 2, we need to show that regular LTFs that are far from monotone must have a large number of reasonably large negative weights. Using this lemma, we obtain the following bound on the magnitude of the negative weights of regular LTFs that are far from monotone.

PROPOSITION 29. *Fix $n \geq 1$ and $\epsilon > 0$. Let $f: \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a τ -regular LTF with weights w_1, \dots, w_n that is ϵ -far from monotone. Assume $\sum_{i \in [n]} w_i^2 = 1$ and $\tau \leq \frac{\epsilon}{4}$. Then the set $N = \{i \in [n] \mid w_i < 0\}$ of indices corresponding to negative weights satisfies $\sum_{i \in N} w_i^2 \geq \frac{\epsilon^2}{256 \ln(8/\epsilon)}$.*

PROOF. Let $g: \{-1, 1\}^n \rightarrow \{-1, 1\}$ be the LTF $g(x) = \text{sgn}(\sum_{i \in [n] \setminus N} w_i x_i - \theta)$ obtained by removing the negative weights of f . Since the function g is monotone,

$$\Pr[f(x) \neq g(x)] \geq \epsilon.$$

The event $f(x) \neq g(x)$ can only occur when $|\sum_{i \in N} w_i x_i| > \left| \sum_{i \in [n] \setminus N} w_i x_i - \theta \right|$. So for any $t > 0$,

$$\begin{aligned} \Pr[f(x) \neq g(x)] &\leq \Pr\left[\left|\sum_{i \in [n] \setminus N} w_i x_i - \theta\right| \leq t\right] + \Pr\left[\left|\sum_{i \in N} w_i x_i\right| > t\right]. \end{aligned}$$

Define $\eta = \sum_{i \in N} w_i^2$. If $\eta > \frac{1}{2}$, then we are done. So assume from now on that $\eta \leq \frac{1}{2}$. Fix $t = \sqrt{2\eta \ln(\frac{8}{\epsilon})}$. By Lemma 5,

$$\begin{aligned} \Pr\left[\left|\sum_{i \in [n] \setminus N} w_i x_i - \theta\right| \leq \sqrt{2\eta \ln(\frac{8}{\epsilon})}\right] &\leq 2\sqrt{\frac{2\eta \ln(\frac{8}{\epsilon})}{1 - \eta}} + \frac{\epsilon}{2} \leq 4\sqrt{\eta \ln(\frac{8}{\epsilon})} + \frac{\epsilon}{2}, \end{aligned}$$

and by the Hoeffding bound,

$$\Pr\left[\left|\sum_{i \in N} w_i x_i\right| > \sqrt{2\eta \ln(\frac{8}{\epsilon})}\right] \leq 2e^{-(2\eta \ln(\frac{8}{\epsilon}))/2\eta} \leq \frac{\epsilon}{4}.$$

Putting all the inequalities together, we obtain the inequality $\frac{\epsilon}{4} \leq 4\sqrt{\eta \ln(\frac{8}{\epsilon})}$, which is satisfied if and only if $\eta \geq \frac{\epsilon^2}{256 \ln(\frac{8}{\epsilon})}$. \square

COROLLARY 30. *Fix $\epsilon > 0$ and $\tau > 0$. There exists $n_0 = n_0(\epsilon, \tau)$ such that for every $n \geq n_0$, if $f: \{-1, 1\}^n \rightarrow \{-1, 1\}$ is a $\frac{\tau}{\sqrt{n}}$ -regular LTF with normalized weights w_1, \dots, w_n ($\sum_{i \in [n]} w_i^2 = 1$) that is ϵ -far from monotone, then the set $N^\dagger = \left\{i \in [n] \mid w_i < -\frac{\epsilon}{\sqrt{512 \ln(\frac{8}{\epsilon})n}}\right\}$ has cardinality $|N^\dagger| \geq \frac{\epsilon^2}{512\tau^2 \ln(\frac{8}{\epsilon})} \cdot n$.*

PROOF. Let n_0 be the minimal positive integer such that $\frac{\tau}{\sqrt{n_0}} < \frac{\epsilon}{4}$. Define $N = \{i \in [n] \mid w_i < 0\}$. By Proposition 29, the sum of the squares of the negative weights is bounded below by $\sum_{i \in N} w_i^2 \geq \frac{\epsilon^2}{256 \ln(\frac{8}{\epsilon})}$. For every element i in $N \setminus N^\dagger$, the weight w_i satisfies $w_i^2 \leq \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})n}$ so $\sum_{i \in N \setminus N^\dagger} w_i^2 \leq |N \setminus N^\dagger| \cdot \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})n} \leq \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})}$ and

$$\begin{aligned} \sum_{i \in N^\dagger} w_i^2 &= \sum_{i \in N} w_i^2 - \sum_{i \in N \setminus N^\dagger} w_i^2 \\ &\geq \frac{\epsilon^2}{256 \ln(\frac{8}{\epsilon})} - \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})} = \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})}. \end{aligned}$$

The regularity of f guarantees that $\sum_{i \in N^\dagger} w_i^2 \leq |N^\dagger| \frac{\tau^2}{n}$ and so $|N^\dagger| \geq \frac{n}{\tau^2} \cdot \frac{\epsilon^2}{512 \ln(\frac{8}{\epsilon})}$. \square

5.3 Proof of Theorem 2

Let f be any $\frac{\tau}{\sqrt{n}}$ -regular LTF that is ϵ -far from monotone. We may assume that $\sum_i w_i^2 = 1$. We modify the bisection algorithm from Algorithm 1 to make the analysis easier. Define

$$c = \frac{\epsilon^2}{512\tau^2 \ln(\frac{8}{\epsilon})} \quad \text{and} \quad \zeta = \frac{\epsilon}{\sqrt{512 \ln(\frac{8}{\epsilon})}}.$$

Let

$$k = \left\lfloor \log(cn) - \max\left\{\log\left(\frac{8\tau}{\zeta}\right), \kappa\left(\frac{1}{8}\right)\right\} \right\rfloor,$$

where κ is as in Lemma 27.

Consider Algorithm 2. Clearly, the algorithm never rejects a monotone function. Let us now assume that f is ϵ -far from monotone. By Corollary 30, the set $N = \{i \in [n] \mid w_i < -\frac{\zeta}{\sqrt{n}}\}$ then has cardinality $|N| \geq cn$.

Algorithm 2 Modified Bisection Algorithm

- 1: Draw $x \in \{-1, 1\}^n$ uniformly at random.
 - 2: Draw $y \in \{-1, 1\}^n$ uniformly at random $8/\epsilon$ times or until $f(x) \neq f(y)$.
 - 3: If no y satisfying the condition $f(x) \neq f(y)$ was found, **accept**.
 - 4: Assume $f(x) = -1$ and $f(y) = 1$. Otherwise, swap x and y .
 - 5: **for** k times **do**
 - 6: Draw $z \in \text{Hybrid}(x, y)$ uniformly at random.
 - 7: If $f(z) = -1$, update $x \leftarrow z$.
 - 8: Otherwise if $f(z) = 1$, update $y \leftarrow z$.
 - 9: **end for**
 - 10: If $|x \cap y| > \frac{3}{2} \cdot \frac{n}{2^k}$, **accept**.
 - 11: Query all inputs in $\text{Hybrid}(x, y)$. **Reject** if a non-monotone edge found; otherwise **accept**.
-

Assume $x \in \{-1, 1\}^n$ is fixed, and y is uniformly sampled from $\{-1, 1\}^n$. First, f is ϵ -far from a constant function, hence, an y satisfying $f(x) \neq f(y)$ will be found with probability at least $\frac{7}{8}$. Next, let $x \triangle y$ be the set of indices where x and y differ. By Chernoff bound, the probability that $|(x \triangle y) \cap N| < cn/4$ is $o(1)$. Thus, with probability at least $\frac{3}{4}$, after Step 4, x and y satisfy $f(x) = -1$, $f(y) = 1$ and the set $S = x \triangle y$ satisfies $|S \cap N| \geq cn/4$.

Let x_ℓ and y_ℓ denote the value of x and y after the ℓ th iteration of the loop in Algorithm 2. In particular, x_0 and y_0 are the inputs x and y after Step 4 as in the previous paragraph. Denote $S_\ell = x_\ell \triangle y_\ell$ and $N_\ell = N \cap S_\ell$. The sets S_0, S_1, S_2, \dots and the sets N_0, N_1, N_2, \dots are randomized bisection processes with the initial sets S and $N \cap S$, respectively. By Lemma 27, with probability at least $\frac{1}{4}$, the sets S_k and N_k satisfy

$$|S_k| < \frac{3}{2} \cdot \frac{|S|}{2^k} \leq O\left(\frac{n}{cn\frac{\zeta}{\tau}}\right) = O\left(\frac{\tau}{c\zeta}\right)$$

and

$$|N_k| > \frac{1}{2} \cdot \frac{|N \cap S|}{2^k} \geq \frac{1}{2} \cdot \frac{cn/4}{cn\frac{\zeta}{8\tau}} \geq \frac{\tau}{\zeta}.$$

In turn, this implies that the sum of the weights with coordinates in N_k satisfies

$$\sum_{i \in N_k} |w_i| \geq |N_k| \cdot \min_{i \in N_k} |w_i| > \frac{\tau}{\zeta} \cdot \frac{\zeta}{\sqrt{n}} = \frac{\tau}{\sqrt{n}} \geq \max_j |w_j|.$$

Let $f_{x,y}$ denote f restricted to the set $\text{Hybrid}(x, y)$, where x and y are as in Step 10 of the algorithm. This function is non-constant since $f_{x,y}(x) = -1$ and $f_{x,y}(y) = 1$. By Proposition 28, $f_{x,y}$ is a non-monotone LTF on $|S_k| = O(\frac{\tau}{c\zeta})$ variables. Then the algorithm rejects in Step 11 after additional $2^{|S_k|} = 2^{\tilde{O}(\tau^3/\epsilon^3)}$ queries.

5.4 Truncated Functions

In Section 5.3, we showed that the bisection algorithm efficiently ϵ -tests regular LTFs for monotonicity, as specified by Theorem 2. However, in the actual lower bounds by Fischer *et al.* [14], and Chen *et al.* [10, 9], *truncated* LTFs are used, as in Definition 7. In this section, we show that if the bisection algorithm can test some class of functions for monotonicity, then it can also test the truncated version of the same class with a modest slow-down.

Towards this goal, we argue that with probability $\Omega_\epsilon(1)$, the bisection algorithm only queries inputs in the middle layers of the cube, i.e., satisfying $\frac{n}{2} - \delta\sqrt{n} \leq |x| \leq \frac{n}{2} + \delta\sqrt{n}$ in the notation of Definition 7. It is easy to modify the parameters of Algorithm 2 in Section 5.3 so that the algorithm uses $O_{\epsilon, \tau, p}(1) + \log n$ queries, always accepts a monotone function, and rejects a non-monotone $\frac{\tau}{\sqrt{n}}$ -regular LTF with probability at least $1 - p$. Combining the two statements, we get an algorithm that ϵ -tests truncated $\frac{\tau}{\sqrt{n}}$ -regular LTFs for monotonicity in $O_{\epsilon, \tau}(\log n)$ queries.

Consider the performance of Algorithm 2 on a function of the form $\text{Truncate}_\delta(f)$. The algorithm does not know the value of δ , but it knows ϵ , the distance from a non-monotone function $\text{Truncate}_\delta(f)$ to the closest monotone function. By Lemma 5, there exists $\beta = \beta(\epsilon) > 0$ such that with probability at least $\frac{\epsilon}{2}$, the input y found on Step 2 of the algorithm satisfies

$$\frac{n}{2} - (\delta - \beta)\sqrt{n} \leq |y| \leq \frac{n}{2} + (\delta - \beta)\sqrt{n}.$$

The input x also satisfies the same estimates with probability $\Omega_\epsilon(1)$.

Let x_ℓ, y_ℓ and S_ℓ be as in Section 5.3. Let also z_ℓ denote the input z on the $(\ell + 1)$ st iteration of the loop, so that either $x_{\ell+1}$ or $y_{\ell+1}$ equals z_ℓ . We consider those executions of the algorithm, in which

$$\max \left\{ \left| |z_\ell \cap S_\ell \cap x_\ell| - \frac{|S_\ell \cap x_\ell|}{2} \right|, \left| |z_\ell \cap S_\ell \setminus x_\ell| - \frac{|S_\ell \setminus x_\ell|}{2} \right| \right\} \leq \frac{\beta(1 - \alpha)}{4} \alpha^\ell \sqrt{n} \quad (23)$$

for all $\ell \leq k$, where $k = \lceil \log(4\sqrt{n}/\beta) \rceil$ and $\alpha = 0.9$ (or any other constant $\frac{1}{\sqrt{2}} < \alpha < 1$).

We first show that (23) is satisfied for all $\ell \leq k$ with probability $\Omega_\epsilon(1)$. By induction, for each ℓ ,

$$\begin{aligned} |S_\ell| &\leq \frac{n}{2^\ell} + \frac{\beta(1 - \alpha)}{2} \sum_{i=0}^{\ell-1} \frac{\alpha^i}{2^{\ell-i-1}} \sqrt{n} \\ &\leq \frac{n}{2^\ell} + \frac{\beta(1 - \alpha)}{2\alpha - 1} \alpha^\ell \sqrt{n} \leq 2 \cdot \frac{n}{2^\ell}, \end{aligned} \quad (24)$$

where the last inequality holds if n is large enough. By the Chernoff-Hoeffding bound, the probability that (23) is satisfied for all $\ell \leq k$ is at least

$$\begin{aligned} \prod_{\ell=0}^k \left(1 - 2 \exp \left(-2 \frac{\beta^2(1-\alpha)^2}{16} \alpha^{2\ell} n \right) \right)^2 \\ \geq \left(\prod_{\ell=0}^{\infty} \left(1 - 2e^{-\Omega_\epsilon((2\alpha^2)^\ell)} \right) \right)^2, \end{aligned}$$

and the infinite product converges.

By induction again, for each $\ell \leq k$,

$$\left| |z_\ell| - \frac{n}{2} \right| < (\delta - \beta)\sqrt{n} + \frac{\beta(1 - \alpha)}{2} \sum_{\ell=0}^{+\infty} \alpha^\ell \sqrt{n} = \left(\delta - \frac{\beta}{2} \right) \sqrt{n}.$$

Also, by (24), $|S_k| \leq \frac{\beta}{2} \sqrt{n}$. Hence, all the inputs queried by the algorithm after the k th iteration of the loop are also in the middle layers of the cube.

5.5 Acknowledgements

The authors thank Sofya Raskhodnikova for communicating the idea of the adaptive bisection algorithm for testing monotonicity.

A.B. is supported by FP7 FET Proactive project QALGO. Part of this work was completed while A.B. was at the University of Latvia. E.B. is supported by an NSERC Discovery grant.

6. REFERENCES

- [1] A. Belovs and E. Blais. Quantum algorithm for monotonicity testing on the hypercube. *Theory of Computing*, 11(16):403–412, 2015.
- [2] P. Berman, S. Raskhodnikova, and G. Yaroslavtsev. L_p -testing. In *Proc. of 46th ACM STOC*, pages 164–173, 2014.
- [3] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.
- [4] A. Bogdanov and M. Safra. Hardness amplification for errorless heuristics. In *Proc. of 48th IEEE FOCS*, pages 418–426, 2007.
- [5] J. Briët, S. Chakraborty, D. García-Soriano, and A. Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012.
- [6] C. Canonne. Open problem for february 2015. Property Testing Review Blog post), 2015.
- [7] D. Chakrabarty and C. Seshadhri. A $o(n)$ monotonicity tester for Boolean functions over the hypercube. In *Proc. of 45th ACM STOC*, pages 411–418, 2013.
- [8] D. Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.
- [9] X. Chen, A. De, R. A. Servedio, and L.-Y. Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proc. of 47th ACM STOC*, pages 519–528, 2015.
- [10] X. Chen, R. A. Servedio, and L.-Y. Tan. New algorithms and lower bounds for monotonicity testing. In *Proc. of 55th IEEE FOCS*, pages 286–295, 2014.
- [11] I. Diakonikolas and R. A. Servedio. Improved approximation of linear threshold functions. *Computational Complexity*, 22(3):623–677, 2013.
- [12] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonicity. In *Proc. of 3rd RANDOM*, pages 97–108. Springer, 1999.
- [13] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of EATCS*, 75:97–126, 2001.
- [14] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proc. of 34th ACM STOC*, pages 474–483, 2002.
- [15] O. Goldreich, S. Goldwasser, E. Lehman, and D. Ron. Testing monotonicity. In *Proc. of 39th IEEE FOCS*, pages 426–435, 1998.
- [16] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.
- [17] S. Khot, D. Minzer, and M. Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *Proc. of 56th IEEE FOCS*, pages 52–58, 2015.
- [18] H. K. Lee. Learning talagrand DNF formulas. In *Proc. of 23rd COLT*, pages 310–311, 2010.
- [19] K. Matulef, R. O’Donnell, R. Rubinfeld, and R. A. Servedio. Testing halfspaces. *SIAM Journal on Computing*, 39(5):2004–2047, 2010.
- [20] E. Mossel and R. O’Donnell. On the noise sensitivity of monotone functions. *Random Structures and Algorithms*, 23(3):333–350, 2003.
- [21] R. O’Donnell and R. A. Servedio. The chow parameters problem. *SIAM Journal on Computing*, 40(1):165–199, 2011.
- [22] R. O’Donnell and K. Wimmer. Approximation by DNF: examples and counterexamples. In *Proc. of 34th ICALP*, pages 195–206, 2007.
- [23] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.
- [24] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.