Adaptive Boolean Monotonicity Testing in Total Influence Time

Deeparnab Chakrabarty
Dartmouth College
deeparnab@dartmouth.edu

C. Seshadhri
University of California, Santa Cruz
sesh@ucsc.edu

Abstract

The problem of testing monotonicity of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ has received much attention recently. Denoting the proximity parameter by ε, the best tester is the non-adaptive $\widetilde{O}(\sqrt{n}/\varepsilon^2)$ tester of Khot-Minzer-Safra (FOCS 2015). Let $\mathbf{I}(f)$ denote the total influence of f. We give an adaptive tester whose running time is $\mathbf{I}(f)\mathrm{poly}(\varepsilon^{-1}\log n)$.

1 Introduction

Consider the boolean hypercube $\{0,1\}^n$, endowed with the coordinate-wise partial order denoted by $\prec$. A function $f : \{0,1\}^n \to \{0,1\}$ is monotone if $\forall x \prec y, f(x) \le f(y)$. Let $d(f,g)$, the distance between two functions f, g, be $|\{x : f(x) \ne g(x)\}|/2^n$. The distance to monotonicity, ε_f, is the minimum distance of f to a monotone g. The problem of monotonicity testing is as follows. Given query access to f and a proximity parameter $\varepsilon > 0$, design a (randomized) procedure that accepts if f is monotone ($\varepsilon_f = 0$), and rejects if f is ε-far from monotone ($\varepsilon_f > \varepsilon$). Both the above guarantees hold with probability at least $2/3$. Such a procedure is called a monotonicity tester. A tester is non-adaptive if the queries made are independent of the values of f, and adaptive otherwise. A tester is one-sided if it accepts monotone functions with probability 1, and two-sided otherwise.

The problem of monotonicity testing over the hypercube has been the subject of much study [Ras99, GGL+00, DGL+99, CS14, CST14, CDST15, KMS15, BB16, CWX17]. The first result was the $O(n/\varepsilon)$ tester by Goldreich et al. and Dodis et al. [GGL+00, DGL+99] (refer to Raskhodnikova's thesis [Ras99]). The authors unearthed a connection to directed isoperimetric theorems to give a $o(n)$ tester [CS14], whose analysis was further improved by Chen et al. [CST14]. In a remarkable result, Khot-Minzer-Safra (henceforth KMS) designed an $\widetilde{O}(\sqrt{n}/\varepsilon^2)$-query non-adaptive, one-sided tester [KMS15]. The key ingredient is a directed analog of Talagrand's isoperimetry theorem [Tal93].

The significance of the $\sqrt{n}$ bound of KMS to f is underscored by a nearly matching lower bound by Fischer et al. for non-adaptive, one-sided testers [FLN+02]. A nearly matching two-sided, non-adaptive lower bound of $\Omega(n^{1/2-\delta})$ for any $\delta > 0$ was showed by Chen et al. [CST14, CDST15] using highly non-trivial techniques of generalized Central Limit Theorems.

In a major advance, Belovs and Blais [BB16] proved the first polynomial *adaptive* (two-sided) lower bound of $\widetilde{\Omega}(n^{1/4})$. This was further improved by Chen et al. [CWX17] to $\widetilde{\Omega}(n^{1/3})$. Both of these are highly non-trivial results, showing that challenges in arguing about adaptive monotonicity testers. Interestingly, Belovs and Blais [BB16] show $O(\log n) + 2^{O(1/\varepsilon^3)}$ time adaptive monotonicity testers for Regular Linear Threshold Functions, which form the hard distribution of the nearly $\Omega(\sqrt{n})$ lower bound of Chen et al. [CST14, CDST15].

This leads to the main open problem in monotonicity testing of Boolean functions: Does adaptivity always help?

1.1 Results

Given our (current) inability to resolve that question, we ask a refined, simpler question. Is there some natural condition of functions under which the $\sqrt{n}$ non-adaptive complexity of KMS can be beaten?

Let $\mathbf{I}(f)$ denote the total influence (also called the average sensitivity) of f. Letting $\mathcal{D}$ be the uniform distribution over all pairs (x, y) at Hamming distance 1, $\mathbf{I}(f) := n \cdot \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq f(y)]$. Our main theorem follows.

Theorem 1.1. *Consider the class of functions with total influence at most I. There exists a one-sided, adaptive monotonicity tester for this class with running time $O(I \cdot \mathrm{poly}(\log n)/\varepsilon^4)$.*

Thus, for low influence functions, the KMS $\sqrt{n}$ bound can be (adaptively) beaten. The non-adaptive lower bound of Fischer et al. [FLN+02] only requires constant influence functions, so Theorem 1.1 shows that adaptivity provably helps for such functions. For $\mathbf{I}(f) \gg \sqrt{n}$, a claim of KMS (explained later) implies the existence of non-adaptive $O(n/\mathbf{I}(f)))$ testers. The tradeoff between Theorem 1.1 and this bound is basically $\sqrt{n}$, the maximum possible influence of a monotone function. We note that all adaptive lower bound constructions have functions with influence $\Theta(\sqrt{n})$ [BB16, CWX17].

2 Preliminaries

We use H_n to denote the standard (undirected) hypercube graph on $\{0, 1\}^n$, where all pairs at Hamming distance 1 are connected. An edge (x, y) of H_n is influential, if $f(x) \neq f(y)$. The number of influential edges is precisely $\mathbf{I}(f) \cdot 2^{n-1}$. An influential edge (x, y) is a violating edge if $x \prec y$, $f(x) = 1$, and $f(y) = 0$. Our tester will perform random walks of H_n. Note that H_n is regular, so this is a symmetric Markov Chain.

We crucially use the central result of KMS, in essence a directed analogue of Talagrand's isoperimetric theorem.

Lemma 2.1. *(Lemma 7.1 in [KMS15], paraphrased) Given any Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ which is ε-far from being monotone, there exists a subgraph $G = (A, B, E)$ of the hypercube and parameters $\sigma \in (0, 1)$, $d \in \mathbb{Z}_{>0}$ such that*

- *Each edge $(a, b) \in E$ with $a \in A$ and $b \in B$ is a violating edge.*
- *$|B| = \sigma \cdot 2^n$.*
- *The degree of each vertex in B is exactly d and the degree of each vertex in A is at most $2d$.*
- *$\sigma^2 d = \Theta(\varepsilon^2/\log^4 n)$.*

We also mention a simpler lemma from KMS which we use.

Lemma 2.2. *(Theorem 9.1 in [KMS15], paraphrased) If $\mathbf{I}(f) > 6\sqrt{n}$, then a constant fraction of influential edges are violating edges.*

When $\mathbf{I}(f) > 6\sqrt{n}$, one can find a violation in time $O(n/\mathbf{I}(f)) = O(\mathbf{I}(f))$ by simply sampling random edges. Thus, we will focus on the case $\mathbf{I}(f) \leq 6\sqrt{n}$.

Idea. Our analysis is short and follows the analysis of the KMS tester. The tester of KMS can be throught of as performing random walks on the *directed* hypercube with orientation corresponding to the partial order. Their analysis lower bounds the probability of encountering a violation. Our insight is that one can perform an analogous analysis for walks on the *undirected* hypercube H_n. Suppose we perform an ℓ-step random walk (on H_n) from x that ends at y. If $f(x) \neq f(y)$, then the walk clearly passed through an influential edge. The power of adaptivity is that we can find such an influential edge through binary search. This idea of using binary search is also present in the algorithm of Belovs and Blais [BB16] to adaptively test regular linear threshold functions. We bound the probability that this influential edge is a violation. Our insight is that, by setting the length ℓ appropriately, this probability can be lower bounded by (essentially) $1/\mathbf{I}(f)$.

3 Tester

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function over the hypercube with total influence $\mathbf{I}(f)$.

Input: A Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ and a parameter $\varepsilon \in (0,1)$

1. Choose $k \in_R \{0,1,2,\ldots,\lceil \log n \rceil\}$ uniformly at random. Set $\ell := 2^k$.
2. Choose $x \in \{0,1\}^n$ uniformly at random.
3. Perform an ℓ-length random walk p on H_n starting from x to reach $y \in \{0,1\}^n$.
4. If $f(x) \neq f(y)$:

    (a) Perform binary search on p to find an influential edge $(u,v) \in p$.

    (b) REJECT if (u,v) is a monotonicity violation.

5. If not REJECTED so far, then ACCEPT.

Figure 1: **Adaptive Monotonicity Tester for Boolean Functions**

Theorem 1.1 follows directly from the following.

Theorem 3.1. *Assume $\mathbf{I}(f) \leq 6\sqrt{n}$. If f is ε-far from being monotone, then Algorithm 1 rejects with probability $\Omega\left(\frac{\varepsilon^4}{\mathbf{I}(f)\log^9 n}\right)$.*

Definition 3.2. *Given a positive integer ℓ, we call a vertex $x \in \{0,1\}^n$ ℓ-sticky if an ℓ-length random walk from x on H_n contains no influential edges with probability $\geq 1/2$. A vertex is called non-ℓ-sticky otherwise. An edge is ℓ-sticky if both endpoints are ℓ-sticky.*

The following simple monotonicity property follows from the definition.

Observation 3.3. *If x is ℓ-sticky and $\ell' < \ell$, then x is ℓ'-sticky as well.*

Lemma 3.4. *The fraction of non-ℓ-sticky vertices of a hypercube is at most $\frac{2\ell \cdot \mathbf{I}(f)}{n}$.*

Proof. Given $x \in \{0,1\}^n$ and a positive integer $\ell > 0$, define the random variable $Z_{x,\ell}$ that is the number of influential edges in a random walk of length ℓ starting from x. Therefore, x is ℓ-non-sticky iff $\Pr[Z_{x,\ell} > 0] > 1/2$. Let N denote the set of ℓ-non-sticky vertices.

Since $Z_{x,\ell}$ is non-negative and integer valued we get $\Pr[Z_{x,\ell} > 0] \le \mathbf{E}[Z_{x,\ell}]$.

$$|N|/2^n < \frac{2}{2^n} \sum_{x \in N} \Pr[Z_{x,\ell} > 0] < \frac{2}{2^n} \sum_{x \in \{0,1\}^n} \Pr[Z_{x,\ell} > 0] \le \frac{2}{2^n} \sum_{x \in \{0,1\}^n} \mathbf{E}[Z_{x,\ell}] \qquad (1)$$

The RHS above is precisely twice the expected number of influential edges encountered in an ℓ-length random walk starting from the uniform distribution on H_n. Let $\mathcal{P}_\ell$ denote the uniform distribution on ℓ-length paths in H_n. For $p \sim \mathcal{P}_\ell$, p_t denotes the tth edge in p, and let $\chi(e)$ be the indicator for edge e being influential. The RHS of (1) is equal to $2\mathbf{E}_{p \sim \mathcal{P}_\ell}[\sum_{t \le \ell} \chi(p_t)] = 2 \sum_{t \le \ell} \mathbf{E}_{p \sim \mathcal{P}_\ell}[\chi(p_t)]$. Since the uniform distribution is stationary for random walks on H_n, the distribution induced on p_t is the uniform distribution on edges in H_n. Thus, $\mathbf{E}_{p \sim \mathcal{P}_\ell}[\chi(p_t)] = \mathbf{I}(f)/n$ and the RHS of (1) is $2\ell \cdot \mathbf{I}(f)/n$. $\qquad \square$

For any integer $\ell > 0$, let F_ℓ be the set of ℓ-sticky violating edges. That is,

$$F_\ell := \{(x,y) \in H_n : (x,y) \text{ is violating and, } x, y \text{ are } \ell\text{-sticky}\}$$

Lemma 3.5. *If ℓ is the length of the random walk chosen in Step 1, then Algorithm 1 rejects with probability* $\Omega\left(\frac{\ell}{n} \cdot \frac{|F_\ell|}{2^n}\right)$.

Proof. Fix an edge $(u,v) \in F_\ell$. Let $p = (e_1, \ldots, e_\ell)$ be the edges of a random walk p. Each edge e_t for $1 \le t \le \ell$ is a uniform random edge of H_n. Therefore, for any fixed edge (u,v), $\Pr[e_t = (u,v)] = \frac{2}{n2^n}$.

Now, given $1 \le t \le \ell$, define $\mathcal{E}_{u,v}^t$ to be the event that the tth edge of p is (u,v) *and* no other edge of p is influential. Since (u,v) is itself influential, $\mathcal{E}_{u,v}$ is the disjoint union $\vee_{t=1}^\ell \mathcal{E}_{u,v}^t$. Furthermore, for two distinct edges (u,v) and (u',v') in F_ℓ, the events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{u',v'}$ are disjoint. Observe that

$$\Pr[\text{Algorithm 1 rejects given } \ell] \ge \Pr[\bigvee_{(u,v) \in F_\ell} \mathcal{E}_{u,v}] = \sum_{(u,v) \in F_\ell} \Pr[\mathcal{E}_{u,v}] \qquad (2)$$

The equality follows since the events are mutually exclusive. The inequality follows since if $\mathcal{E}_{u,v}$ occurs then the end points of p must have differing values and binary search on p will return the violation (u,v).

Consider the event $\mathcal{E}_{u,v}^t$. For this event to occur, e_t must be (u,v). Consider the conditional probability $\Pr[\mathcal{E}_{u,v}^t \mid e_t = (u,v)]$. Let $\mathcal{F}_u$ be the event that a $(t-1)$-length random walk from u contains no influential edges, and let $\mathcal{F}_v$ be the event that an *independent* $(\ell - t)$-length random walk from v contains no influential edges.

Claim 3.6. $\Pr[\mathcal{E}_{u,v}^t \mid e_t = (u,v)] = \Pr[\mathcal{F}_u \wedge \mathcal{F}_v] = \Pr[\mathcal{F}_u] \cdot \Pr[\mathcal{F}_v]$

Proof. Conditioned on $e_t = (u,v)$, the distribution of the first $(t-1)$ steps of the random walk is the uniform distribution of $(t-1)$-length paths that end at u. This is the same distribution of the $(t-1)$-length random walk starting from u. The distribution of the last $(\ell - t)$ steps, conditioned on $e_t = (u,v)$, is the $(\ell - t)$-length random walk starting from v. $\qquad \square$

Since (u, v) is an ℓ-sticky edge, by Obs 3.3 and Definition 3.2, $\Pr[\mathcal{F}_u] \geq 1/2$ and $\Pr[\mathcal{F}_v] \geq 1/2$. The proof is completed by plugging the following bound into (2).

$$\Pr[\mathcal{E}_{u,v}] = \sum_{t=1}^{\ell} \Pr[\mathcal{E}_{u,v}^t] = \sum_{t=1}^{\ell} \Pr[e_t = (u,v)] \cdot \Pr[\mathcal{E}_{u,v}^t \mid e_t = (u,v)] = \frac{2}{n2^n} \sum_{t=1}^{\ell} \Pr[\mathcal{E}_{u,v}^t \mid e_t = (u,v)]$$
$$\geq \frac{\ell}{4n2^n}$$

$\square$

We complete the proof of Theorem 3.1.

Proof of Theorem 3.1. Let $G = (A, B, E), \sigma, d$ be as in Lemma 2.1.

First, we take care of a corner case. Suppose $\sigma < 100/\sqrt{n}$. Since $\sigma^2 d = \Omega(\varepsilon^2/\log^4 n)$, we get $\sigma d = \Omega(\varepsilon^2 \sqrt{n}/\log^4 n)$. Since the number of violating edges is $\geq |E| = \sigma d 2^n$ and since each violating edge is influential, we get that $\mathbf{I}(f) \geq |E|/2^n = \Omega(\varepsilon^2 \sqrt{n}/\log^4 n)$. With probability $1/\log n$, Algorithm 1 sets $\ell = 1$, in which case the tester basically checks whether a uniform random edge in H_n is a violation. Conditioned on setting $\ell = 1$, the rejection probability is $\Omega(2|E|/n2^n) = \Omega(\varepsilon^2/\sqrt{n}\log^4 n) = \Omega(\varepsilon_f^4/\mathbf{I}(f)\log^8 n)$. Thus, in the case $\sigma < 100/\sqrt{n}$, we have proved Theorem 3.1.

Henceforth, assume that $\sigma \geq 100/\sqrt{n}$. Since $\mathbf{I}(f) \leq 6\sqrt{n}$, we deduce that $\frac{n\sigma}{16 \cdot \mathbf{I}(f)} > 1$. Thus, there exists a non-negative power of 2 (call it ℓ^*) such that $\frac{\sigma}{16} < \frac{\ell^* \cdot \mathbf{I}(f)}{n} \leq \frac{\sigma}{8}$.

Let A' and B' be the subset of A and B that are ℓ^*-sticky. Let $E' \subseteq E$ be the edges with end points in A' and B'. Note that any edge of $E' \subseteq F_{\ell^*}$. By Lemma 3.4, we get that the fraction of ℓ^*-non-sticky nodes is at most $2\ell^* \cdot \mathbf{I}(f)/n \leq \sigma/4$. Since the degree of any node in G in Lemma 2.1 is $\leq 2d$, we get

$$|F_{\ell^*}| \geq |E'| \geq |E| - (2d) \cdot \frac{\sigma \cdot 2^n}{4} = \frac{\sigma d 2^n}{2}.$$

The probability the algorithm chooses $\ell = \ell^*$ is $1/\log n$. Lemma 3.5 gives us

$$\Pr[\text{Algorithm rejects}] \geq \frac{1}{\log n} \cdot \Pr[\text{Algorithm rejects}|\ell = \ell^*]$$
$$\geq \frac{1}{\log n} \cdot \left(\frac{\ell^*}{n} \cdot \frac{|F_{\ell^*}|}{2^n} \right) \qquad \text{(by Lemma 3.5)}$$
$$\geq \frac{1}{\log n} \cdot \frac{\ell^*}{n} \cdot \frac{\sigma d}{2}$$
$$\geq \frac{1}{\mathbf{I}(f)} \cdot \left(\frac{\sigma^2 d}{32 \log n} \right) \qquad \text{(plugging } \ell^* \geq \sigma n/(16 \cdot \mathbf{I}(f)))$$
$$\geq \frac{1}{\mathbf{I}(f)} \cdot \frac{\varepsilon^2}{\log^5 n} \qquad \text{(by Lemma 2.1)}$$

$\square$

References

[BB16]   Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 1021–1032, 2016. 1, 2, 3

[CDST15]   Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $O(n^{1/2})$ non-adaptive queries. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 519–528, 2015. 1

[CS14]   Deeparnab Chakrabarty and C. Seshadhri. An o(n) monotonicity tester for boolean functions over the hypercube. *SIAM Journal on Computing*, 2014. 1

[CST14]   Xi Chen, Rocco A. Servedio, and Li-Yang. Tan. New algorithms and lower bounds for monotonicity testing. In *Proceedings of Foundations of Computer Science (FOCS)*, pages 286–295, 2014. 1

[CWX17]   Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond talagrand: New lower bounds for testing monotonicity and unateness. In *ACM Symposium on Theory of Computation (STOC)*, 2017. 1, 2

[DGL+99]   Yevgeny Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. *Proceedings of the 3rd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 97–108, 1999. 1

[FLN+02]   Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, and Ronitt Rubinfeld. Monotonicity testing over general poset domains. *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 474–483, 2002. 1, 2

[GGL+00]   Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samordinsky. Testing monotonicity. *Combinatorica*, 20:301–337, 2000. 1

[KMS15]   Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric type theorems. In *IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2015. 1, 2

[Ras99]   Sofya Raskhodnikova. Monotonicity testing. *Masters Thesis, MIT*, 1999. 1

[Tal93]   Michel Talagrand. Isoperimetry, logarithmic sobolev inequalities on the discrete cube, and margulis graph connectivity theorem. *Geom. Func. Anal.*, 3(3):295–314, 1993. 1