Learning Circuits with Few Negations

Eric Blais[1], Clément L. Canonne[2], Igor C. Oliveira[2], Rocco A. Servedio[2], and Li-Yang Tan[2]

1 **University of Waterloo**
200 University Avenue West, Waterloo ON, Canada
eric.blais@uwaterloo.ca

2 **Columbia University**
500 W 120th Street, New York NY, USA
{ccanonne,oliveira,rocco,liyang}@cs.columbia.edu

Abstract

Monotone Boolean functions, and the monotone Boolean circuits that compute them, have been intensively studied in complexity theory. In this paper we study the structure of Boolean functions in terms of the minimum number of negations in any circuit computing them, a complexity measure that interpolates between monotone functions and the class of all functions. We study this generalization of monotonicity from the vantage point of learning theory, establishing nearly matching upper and lower bounds on the uniform-distribution learnability of circuits in terms of the number of negations they contain. Our upper bounds are based on a new structural characterization of negation-limited circuits that extends a classical result of A. A. Markov. Our lower bounds, which employ Fourier-analytic tools from hardness amplification, give new results even for circuits with no negations (i.e. monotone functions).

1998 ACM Subject Classification I.2.6 Learning, F.2.2 Nonnumerical Algorithms and Problems, G.1.2 Approximation, G.3 Probability and Statistics

Keywords and phrases Boolean functions, monotonicity, negations, PAC learning

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2015.512

1 Introduction

A *monotone* Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is one that satisfies $f(x) \leq f(y)$ whenever $x \preceq y$, where $\preceq$ denotes the bitwise partial order on $\{0,1\}^n$. The structural and combinatorial properties of monotone Boolean functions have been intensively studied for many decades, see e.g. [12] for an in-depth survey. Many important results in circuit complexity deal with monotone functions, including celebrated lower bounds on monotone circuit size and monotone formula size (see e.g. [22, 23] and numerous subsequent works).

Monotone functions are also of considerable interest in computational learning theory, in particular with respect to the model of learning under the uniform distribution. In an influential paper, Bshouty and Tamon [6] showed that any monotone Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ can be learned from uniform random examples to error ε in time $n^{O(\sqrt{n}/\varepsilon)}$. They also gave a lower bound, showing that no algorithm running in time 2^{cn} for any $c < 1$ can learn arbitrary monotone functions to accuracy $\varepsilon = 1/(\sqrt{n}\log n)$. (Many other works in learning theory such as [3, 11, 5, 1, 26, 20, 21] deal with learning monotone functions from a range of different perspectives and learning models, but we limit our focus in this paper to learning to high accuracy with respect to the uniform distribution.)

18th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'15) / 19th Int'l Workshop on Randomization and Computation (RANDOM'15).
Editors: Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim; pp. 512–527

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Beyond monotonicity: Inversion complexity, alternations, and Markov's theorem

Given the importance of monotone functions in complexity theory and learning theory, it is natural to consider various generalizations of monotonicity. One such generalization arises from the simple observation that monotone Boolean functions are precisely the functions computed by *monotone Boolean circuits*, i.e. circuits which have only AND and OR gates but no negations. Given this, an obvious generalization of monotonicity is obtained by considering functions computed by Boolean circuits that have a small number of negation gates. The *inversion complexity* of $f\colon \{0,1\}^n \to \{0,1\}$, denoted $I(f)$, is defined to be the minimum number of negation gates in any AND/OR/NOT circuit (with access to constant inputs 0/1) that computes f. We write $\mathcal{C}_t^n$ to denote the class of n-variable Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ that have $I(f) \leq t$.

Another generalization of monotonicity is obtained by starting from an alternate characterization of monotone Boolean functions. A function $f\colon \{0,1\}^n \to \{0,1\}$ is monotone if and only if the value of f "flips" from 0 to 1 at most once as the input x ascends any chain in $\{0,1\}^n$ from 0^n to 1^n. (Recall that a chain of length ℓ is an increasing sequence $(x^1, \ldots, x^\ell)$ of vectors in $\{0,1\}^n$, i.e. for every $j \in \{1, \ldots, \ell-1\}$ we have $x^j \prec x^{j+1}$.) Thus, it is natural to consider a generalization of monotonicity that allows more than one such "flip" to occur. We make this precise with the following notation and terminology: given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ and a chain $X = (x^1, \ldots, x^\ell)$, a position $j \in [\ell - 1]$ is said to be *alternating* with respect to f if $f(x^j) \neq f(x^{j+1})$. We write $A(f, X) \subseteq [\ell - 1]$ to denote the set of alternating positions in X with respect to f, and we let $a(f, X) = |A(f, X)|$ denote its size. We write $a(f)$ to denote the maximum of $a(f, X)$ taken over all chains X in $\{0,1\}^n$, and we say that $f\colon \{0,1\}^n \to \{0,1\}$ is *k-alternating* if $a(f) \leq k$.

A celebrated result of A. A. Markov from 1957 [14] gives a tight quantitative connection between the inversion and alternation complexities defined above:

▶ **Markov's Theorem.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a function which is not identically 0. Then* (i) *if $f(0^n) = 0$, then $I(f) = \lceil \log(a(f) + 1) \rceil - 1$; and* (ii) *if $f(0^n) = 1$, then $I(f) = \lceil \log(a(f) + 2) \rceil - 1$.*

This robustness motivates the study of circuits which contain few negation gates, and indeed such circuits have been studied in complexity theory. Amano and Maruoka [2] have given bounds on the computational power of such circuits, showing that circuits for the clique function which contain fewer than $\frac{1}{6} \log \log n$ many negation gates must have superpolynomial size. More recently, Rossman [24] proved that there exists an explicit monotone function that cannot be computed by fan-in two circuits of logarithmic depth containing less than $\left(\frac{1}{2} - \varepsilon\right) \log n$ negations. Other works have studied the effect of limiting the number of negation gates in formulas [16, 9], bounded-depth circuits [25, 27], and non-deterministic circuits [17]. Another line of work that has received attention lately is the role of monotonicity and negation complexity in cryptography and related areas [8, 10].

In the present work, we study circuits with few negations from the vantage point of computational learning theory, giving both positive and negative results. We observe that some of the recent works mentioned [10, 9] build on techniques introduced in a preliminary version of this paper.

1.2 Our results

We begin by studying the structural properties of functions that are computed or approximated by circuits with few negation gates. In Section 2 we establish the following extension of Markov's theorem:

▶ **Theorem 1.1.** *Let f be a k-alternating Boolean function. Then f can be expressed as $f(x) = h(m_1(x), \ldots, m_k(x))$, where each $m_i(x)$ is monotone and h is either the parity function or its negation. Conversely, any function of this form is k-alternating.*

Theorem 1.1 along with Markov's theorem yields the following characterization of $\mathcal{C}_t^n$:

▶ **Corollary 1.2.** *Every $f \in \mathcal{C}_t^n$ can be expressed as $f = h(m_1, \ldots, m_T)$ where h is either PAR_T or its negation, each $m_i \colon \{0,1\}^n \to \{0,1\}$ is monotone, and $T = O(2^t)$.*

A well-known consequence of Markov's theorem is that every Boolean function is exactly computed by a circuit which has only $\log n$ negation gates, and as we shall see an easy argument shows that every Boolean function is 0.01-approximated by a circuit with $\frac{1}{2}\log n + O(1)$ negations. In Section 2 we note that no significant savings are possible over this easy upper bound:

▶ **Theorem 1.3.** *For almost every function $f \colon \{0,1\}^n \to \{0,1\}$, any Boolean circuit C that 0.01-approximates f must contain $\frac{1}{2}\log n - O(1)$ negations.*

We then turn to our main topic of investigation, the uniform-distribution learnability of circuits with few negations. We use our new extension of Markov's theorem, Theorem 1.1, to obtain a generalization of the Fourier-based uniform-distribution learning algorithm of Bshouty and Tamon [6] for monotone circuits:

▶ **Theorem 1.4.** *There is a uniform-distribution learning algorithm which learns any unknown $f \in \mathcal{C}_t^n$ from random examples to error ε in time $n^{O(2^t\sqrt{n}/\varepsilon)}$.*

We observe that many natural functions are indeed computed by circuits with few negations. As an example, consider the property of undirected graphs that is satisfied by an n-vertex graph G if and only if G contains a triangle but does not contain a cycle of size $\log n$. Clearly, this property is non-monotone. However, it is easy to see that it can be represented by a Boolean function $f \colon \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ that is computed by a circuit with a single negation. Our positive result implies that learning such properties does not take much more time than learning monotone properties.[1]

Theorem 1.4 immediately leads to the following question: can an even faster learning algorithm be given for circuits with t negations, or is the running time of Theorem 1.4 essentially the best possible? Interestingly, prior to our work a matching lower bound for Theorem 1.4 was not known even for the special case of monotone functions (corresponding to $t = 0$). As mentioned earlier, Bshouty and Tamon proved that to achieve accuracy $\varepsilon = 1/(\sqrt{n}\log n)$ any learning algorithm needs time $\omega(2^{cn})$ for any $c < 1$ (see Claim 3.13 for a slight sharpening of this statement). For larger values of ε, though, the strongest previous lower bound was due to Blum, Burch and Langford [5]. Their Theorem 10 implies that any membership-query algorithm that learns monotone functions to error $\varepsilon < \frac{1}{2} - c$ (for any $c > 0$) must run in time $2^{\Omega(\sqrt{n})}$ (in fact, must make at least this many membership queries). However, this lower bound does not differentiate between the number of membership queries required to learn to high accuracy versus "moderate" accuracy – say, $\varepsilon = 1/n^{1/10}$ versus $\varepsilon = 1/10$. Thus the following question was unanswered prior to the current paper: what is

[1] In contrast to the robustness we show in the learning setting, there are natural computational problems whose complexity changes drastically with the addition of a single negation gate. For instance, checking if a monotone circuit is non-constant is trivial. Nevertheless, it is possible to prove that the same computational problem for circuits with a single negation gate admits polynomial time algorithms if and only if $\mathsf{P} = \mathsf{NP}$.

the best lower bound that can be given, both as a function of n and ε, on the complexity of learning monotone functions to accuracy ε?

We give a fairly complete answer to this question, providing a lower bound as a function of n, ε and t on the complexity of learning circuits with t negations. Our lower bound essentially matches the upper bound of Theorem 1.4, and is thus simultaneously essentially optimal in all three parameters n, ε and t for a wide range of settings of ε and t. Our lower bound result is the following:

▶ **Theorem 1.5.** *For any $t \leq \frac{1}{28}\log n$ and any $\varepsilon \in [1/n^{1/12}, 1/2 - c]$, $c > 0$, any membership-query algorithm that learns any unknown function $f \in \mathcal{C}_t^n$ to error ε must make $2^{\Omega(2^t\sqrt{n}/\varepsilon)}$ membership queries.*

We note that while our algorithm uses only uniform random examples, our lower bound holds even for the stronger model in which the learning algorithm is allowed to make arbitrary membership queries on points of its choosing.

Theorem 1.5 is proved using tools from the study of hardness amplification. The proof involves a few steps. We start with a strong lower bound for the task of learning to high accuracy the class of balanced monotone Boolean functions (reminiscent of the lower bound obtained by Bshouty and Tamon). Then we combine hardness amplification techniques and results on the noise sensitivity of monotone functions in order to get stronger and more general lower bounds for learning monotone Boolean functions to moderate accuracy. Finally, we use hardness amplification once more to lift this result into a lower bound for learning circuits with few negations to moderate accuracy. An ingredient employed in this last stage is to use a k-alternating combining function which "behaves like" the parity function on (roughly) k^2 variables; this is crucial in order for us to obtain our essentially optimal final lower bound of $2^{\Omega(2^t\sqrt{n}/\varepsilon)}$ for circuits with t negations. These results are discussed in more detail in Section 3.2.

Lastly, we mention an interesting research direction left unanswered by our results. Specifically, we focus in this work on the uniform-distribution learnability to *high accuracy*, i.e. when the error parameter ε is thought of as "small" (or at least bounded away from $1/2$). While we provide almost optimal bounds for this regime, the complexity of *weakly* learning circuits with negations – that is obtaining inverse-polynomial advantage over random guessing – remains open. As a concrete question, is there an *efficient* algorithm that learns circuits with a single negation with error at most $1/2 - \Omega(1/n^c)$ for some $c > 0$? (Note that the analogue question for monotone circuits is well-understood [5, 1, 21].)

2 Structural facts about computing and approximating functions with low inversion complexity

2.1 An extension of Markov's theorem

We begin with the proof of our new extension of Markov's theorem. For any $A \subseteq \{0,1\}^n$ let $\mathbf{1}[A]\colon \{0,1\}^n \to \{0,1\}$ be the characteristic function of A. For $f\colon \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$, we write $a_f(x)$ to denote

$$a_f(x) \stackrel{\text{def}}{=} \max\{a(f,X) : X \text{ is a chain that starts at } x\},$$

and note that $a(f) = \max_{x\in\{0,1\}^n}\{a_f(x)\} = a_f(0^n)$. For $0 \leq \ell \leq a(f)$ let us write S_ℓ^f to denote $S_\ell^f \stackrel{\text{def}}{=} \{x \in \{0,1\}^n : a_f(x) = \ell\}$, and let $T_\ell^f \stackrel{\text{def}}{=} S_0^f \cup \cdots \cup S_\ell^f$. We note that $S_0^f, \ldots, S_{a(f)}^f$ partition the set of all inputs: $S_i^f \cap S_j^f = \emptyset$ for all $i \neq j$, and $T_{a(f)}^f = S_0^f \cup \cdots \cup S_{a(f)}^f = \{0,1\}^n$.

We will need the following simple observation:

▶ **Observation 2.1.** *Fix any f and any $x \in \{0,1\}^n$. If $x \in S^f_\ell$ and $y \succ x$ then $y \in S^f_{\ell'}$ for some $\ell' \leq \ell$. Furthermore, if $f(y) \neq f(x)$ then $\ell' < \ell$.*

▶ **Theorem 1.1.** *(Restated) Fix $f\colon \{0,1\}^n \to \{0,1\}$ and let $k \stackrel{\text{def}}{=} a(f)$. Then f can be expressed as $f = h\big(\mathbf{1}[T^f_0], \ldots, \mathbf{1}[T^f_{k-1}]\big)$, where*
(i) *the functions $\mathbf{1}[T^f_\ell]$ are monotone for all $0 \leq \ell \leq k$,*
(ii) *$h\colon \{0,1\}^k \to \{0,1\}$ is PAR_k if $f(0^n) = 0$ and $\neg\,\mathsf{PAR}_k$ if $f(0^n) = 1$,*
and $\mathsf{PAR}_k(x) = x_1 \oplus \cdots \oplus x_k$ is the parity function on k variables. Conversely, for any monotone Boolean functions $m_1, \ldots, m_k$, any Boolean function of the form $h(m_1, \ldots, m_k)$ is k-alternating.

Proof. Claim 1 follows immediately from Observation 2.1 above. The proof of 2 is by induction on k. In the base case $k = 0$, we have that f is a constant function and the claim is immediate.

For the inductive step, suppose that the claim holds for all functions f' that have $a(f') \leq k-1$. We define $f'\colon \{0,1\}^n \to \{0,1\}$ as $f' = f \oplus \mathbf{1}[S^f_k]$. Observation 2.1 implies that $S^{f'}_\ell = S^f_\ell$ for all $0 \leq \ell \leq k-2$ and $S^{f'}_{k-1} = S^f_{k-1} \cup S^f_k$, and in particular, $a(f) = k-1$. Therefore we may apply the inductive hypothesis to f' and express it as $f' = h'\big(\mathbf{1}[T^{f'}_0], \ldots, \mathbf{1}[T^{f'}_{k-2}]\big)$. Since $T^{f'}_\ell = T^f_\ell$ for $0 \leq \ell \leq k-2$, we may use this along with the fact that $\mathbf{1}[S^f_k] = \neg\,\mathbf{1}[T^f_{k-1}]$ to get:

$$f = f' \oplus \mathbf{1}[S^f_k] = h'\big(\mathbf{1}[T^{f'}_0], \ldots, \mathbf{1}[T^{f'}_{k-2}]\big) \oplus \neg\,\mathbf{1}[T^f_{k-1}] = h'\big(\mathbf{1}[T^f_0], \ldots, \mathbf{1}[T^f_{k-2}]\big) \oplus \neg\,\mathbf{1}[T^f_{k-1}]$$

and the inductive hypothesis holds (note that $0^n \in S^f_k$).

The converse is easily verified by observing that any chain in $\{0,1\}^n$ can induce at most $k+1$ possible vectors of values for $(m_1, \ldots, m_k)$ because of their monotonicity. ◀

Theorem 1.1 along with Markov's theorem immediately yields the following corollary:

▶ **Corollary 1.2.** *Every $f \in \mathcal{C}^n_t$ can be expressed as $f = h(m_1, \ldots, m_T)$ where h is either PAR_T or its negation, each $m_i\colon \{0,1\}^n \to \{0,1\}$ is monotone, and $T = O(2^t)$.*

2.2 Approximation

As noted earlier, Markov's theorem implies that every n-variable Boolean function can be exactly computed by a circuit with (essentially) $\log n$ negations (since $a(f) \leq n$ for all f). If we set a less ambitious goal of *approximating* Boolean functions (say, having a circuit correctly compute f on a $1-\varepsilon$ fraction of all 2^n inputs), can significantly fewer negations suffice?

We first observe that every Boolean function f is ε-close (with respect to the uniform distribution) to a function f' that has $a(f') \leq O(\sqrt{n \log 1/\varepsilon})$. The function f' is obtained from f simply by setting $f'(x) = 0$ for all inputs x that have Hamming weight outside of $[n/2 - O(\sqrt{n \log 1/\varepsilon}), n/2 + O(\sqrt{n \log 1/\varepsilon})]$; a standard Chernoff bound implies that f and f' disagree on at most $\varepsilon 2^n$ inputs. Markov's theorem then implies that the inversion complexity $I(f')$ is at most $\frac{1}{2}(\log n + \log\log \frac{1}{\varepsilon}) + O(1)$. Thus, every Boolean function can be approximated to high accuracy by a circuit with only $\frac{1}{2}\log n + O(1)$ negations.

We now show that this upper bound is essentially optimal: for almost every Boolean function, any 0.01-approximating circuit must contain at least $\frac{1}{2}\log n - O(1)$ negations. To

prove this, we recall the notion of the *total influence* of a Boolean function f: this is

$$\mathbf{Inf}[f] = \sum_{i=1}^{n} \mathbf{Inf}_i[f], \quad \text{where} \quad \mathbf{Inf}_i[f] = \mathbf{Pr}_{x\in\{0,1\}^n}[f(x) \neq f(x^{\oplus i})]$$

and $x^{\oplus i}$ denotes x with its i-th coordinate flipped. The total influence of f is easily seen to equal αn, where $\alpha \in [0,1]$ is the fraction of all edges $e = (x, x')$ in the Boolean hypercube that are bichromatic, i.e. have $f(x) \neq f(x')$. In Appendix A.1 we prove the following lemma:

▶ **Lemma 2.2.** *Suppose* $f\colon \{0,1\}^n \to \{0,1\}$ *is such that* $\mathbf{Inf}[f] = \Omega(n)$. *Then* $a(f) = \Omega(\sqrt{n})$.

It is easy to show that a random function has influence $\frac{n}{2}(1 - o(1))$ with probability $1 - 2^{-n}$. Given this, Claim 2.2, together with the elementary fact that whenever f' is ε-close to f then $|\mathbf{Inf}(f') - \mathbf{Inf}(f)| \leq 2\varepsilon n$, directly yields the following:

▶ **Theorem 1.3.** *With probability* $1 - 2^{-n}$, *any* 0.01*-approximator* f' *for a random function* f *must have inversion complexity* $I(f') \geq \frac{1}{2}\log n - O(1)$.

▶ Remark. The results in this section (together with simple information-theoretic arguments showing that random functions are hard to learn) imply that one cannot expect to have a learning algorithm (even to constant accuracy) for the class $\mathcal{C}^n_{\frac{1}{2}\log n + O(1)}$ of circuits with $\frac{1}{2}\log n + O(1)$ negations in time significantly better than 2^n. As we shall see in Section 3.1, for any fixed $\delta > 0$ it *is* possible to learn $\mathcal{C}^n_{(\frac{1}{2}-\delta)\log n}$ to accuracy $1 - \varepsilon$ in time $2^{\tilde{O}(n^{1-\delta})/\varepsilon}$.

3 Learning circuits with few negations

3.1 A learning algorithm for $\mathcal{C}^n_t$

We sketch the learning algorithm and analysis of Bshouty and Tamon [6]; using the results from Section 2 our Theorem 1.4 will follow easily from their approach. Our starting point is the simple observation that functions with good "Fourier concentration" can be learned to high accuracy under the uniform distribution simply by estimating all of the low-degree Fourier coefficients. This fact, established by Linial, Mansour and Nisan, is often referred to as the "Low-Degree Algorithm:"

▶ **Theorem 3.1** (Low-Degree Algorithm ([13])). *Let* $\mathcal{C}$ *be a class of Boolean functions such that for* $\varepsilon > 0$ *and* $\tau = \tau(\varepsilon, n)$,

$$\sum_{|S|>\tau} \widehat{f}(S)^2 \leq \varepsilon$$

for any $f \in \mathcal{C}$. *Then* $\mathcal{C}$ *can be learned from uniform random examples in time* $\mathrm{poly}(n^\tau, 1/\varepsilon)$.

Using the fact that every monotone function $f\colon \{0,1\}^n \to \{0,1\}$ has total influence $\mathbf{Inf}(f) \leq \sqrt{n}$, and the well-known Fourier expression $\mathbf{Inf}(f) = \sum_S \widehat{f}(S) \cdot |S|^2$ for total influence, a simple application of Markov's inequality let Bshouty and Tamon show that every monotone function f has

$$\sum_{|S|>\sqrt{n}/\varepsilon} \widehat{f}(S)^2 \leq \varepsilon.$$

Together with Theorem 3.1, this gives their learning result for monotone functions.

Armed with Corollary 1.2, it is straightforward to extend this to the class $\mathcal{C}_t^n$. Corollary 1.2 and a union bound immediately give that every $f \in \mathcal{C}_t^n$ has $\mathbf{Inf}(f) \leq O(2^t)\sqrt{n}$, so the Fourier expression for influence and Markov's inequality give that

$$\sum_{|S|>O(2^t)\sqrt{n}/\varepsilon} \widehat{f}(S)^2 \leq \varepsilon$$

for $f \in \mathcal{C}_t^n$. Theorem 1.4 follows immediately using the Low-Degree Algorithm.

An immediate question is whether this upper bound on the complexity of learning $\mathcal{C}_t^n$ is optimal; we give an affirmative answer in the next subsection.

3.2 Lower bounds for learning

As noted in the introduction, we prove information-theoretic lower bounds against learning algorithms that make a limited number of membership queries. We start by establishing a new lower bound on the number of membership queries that are required to learn *monotone* functions to high accuracy, and then build on this to provide a lower bound for learning $\mathcal{C}_t^n$. Our query lower bounds are essentially tight, matching the upper bounds (which hold for learning from uniform random examples) up to logarithmic factors in the exponent.

We first state the results; the proofs are deferred to Section 3.2.1. We say that a Boolean function f is *balanced* if $\mathbf{Pr}_x[f(x) = 0] = \mathbf{Pr}_x[f(x) = 1] = 1/2$.

▶ **Theorem 3.2.** *There exists a class $\mathcal{H}_n$ of balanced n-variable monotone Boolean functions such that for any $\varepsilon \in [\frac{1}{n^{1/6}}, 1/2 - c]$, $c > 0$, learning $\mathcal{H}_n$ to accuracy $1 - \varepsilon$ requires $2^{\Omega(\sqrt{n}/\varepsilon)}$ membership queries.*

This immediately implies the following corollary, which essentially closes the gap in our understanding of the hardness of learning monotone functions:

▶ **Corollary 3.3.** *For any $\varepsilon = \Omega(1/n^{1/6})$ bounded away from $1/2$, learning n-variable monotone functions to accuracy $1 - \varepsilon$ requires $2^{\tilde{\Theta}(\sqrt{n})/\varepsilon}$ queries.*

Using this class $\mathcal{H}$ as a building block, we obtain the following hardness of learning result for the class of k-alternating functions:

▶ **Theorem 3.4.** *For any function $k\colon \mathbb{N} \to \mathbb{N}$, there exists a class $\mathcal{H}^{(k)}$ of balanced $k = k(n)$-alternating n-variable Boolean functions such that, for any n sufficiently large and $\varepsilon > 0$ such that (i) $2 \leq k < n^{1/14}$, and (ii) $k^{7/3}/n^{1/6} \leq \varepsilon \leq \frac{1}{2} - c$, learning $\mathcal{H}^{(k)}$ to accuracy $1 - \varepsilon$ requires $2^{\Omega(k\sqrt{n}/\varepsilon)}$ membership queries.*

(We note that the tradeoff between the ranges of k and ε that is captured by condition (ii) above seems to be inherent to our approach and not a mere artifact of the analysis; see Observation 3.16.) This theorem immediately yields the following:

▶ **Corollary 3.5.** *Learning the class of k-alternating functions to accuracy $1 - \varepsilon$ in the uniform-distribution membership-query model requires $2^{\Omega(k\sqrt{n}/\varepsilon)}$ membership queries, for any $k = O(n^{1/28})$ and $\varepsilon \in [1/n^{1/12}, \frac{1}{2} - c]$.*

▶ **Corollary 3.6.** *For $t \leq \frac{1}{28}\log n$, learning $\mathcal{C}_t^n$ to accuracy $1 - \varepsilon$ requires $2^{\Omega(2^t\sqrt{n}/\varepsilon)}$ membership queries, for any $\varepsilon \in [2^{7t/3}/n^{1/6}, \frac{1}{2} - c]$.*

3.2.1 Proofs

We require the following standard notion of *composition* for two functions f and g:

▶ **Definition 3.7** (Composition). For $f\colon \{0,1\}^m \to \{0,1\}$ and $g\colon \{0,1\}^r \to \{0,1\}$, we denote by $g \otimes f$ the Boolean function on $n = mr$ inputs defined by

$$(g \otimes f)(x) \stackrel{\text{def}}{=} g(\underbrace{f,\ldots,f}_{r})(x) = g(f(x_1,\ldots,x_m),\ldots,f(x_{(r-1)m+1},\ldots,x_{rm}))$$

Similarly, for any $g\colon \{0,1\}^r \to \{0,1\}$ and $\mathcal{F}_m$ a class of Boolean functions on m variables, we let

$$g \otimes \mathcal{F}_m = \{\, g \otimes f \,:\, f \in \mathcal{F}_m \,\}$$

and $g \otimes \mathcal{F} = \{g \otimes \mathcal{F}_m\}_{m \geq 1}$.

Overview of the arguments. Our approach is based on hardness amplification. In order to get our lower bound against learning k-alternating functions, we (a) start from a lower bound ruling out very high-accuracy learning of *monotone* functions; (b) use a suitable monotone combining function to get an XOR-like hardness amplification, yielding a lower bound for learning (a subclass of) monotone functions to moderate accuracy; (c) repeat this approach on this subclass with a different (now k-alternating) combining function to obtain our final lower bound, for learning k-alternating functions to moderate accuracy.

$$\underbrace{\left[\begin{array}{c}\text{high-accuracy}\\ \text{monotone}\end{array}\right]}_{\text{(a)}} \xrightarrow[\text{monotone}]{\bigotimes\text{-like}} \underbrace{\left[\begin{array}{c}\text{moderate accuracy}\\ \text{monotone}\end{array}\right]}_{\text{(b)}} \xrightarrow[k\text{-alternating}]{\bigotimes\text{-like}} \underbrace{\left[\begin{array}{c}\text{moderate accuracy}\\ k\text{-alternating}\end{array}\right]}_{\text{(c)}} \quad (1)$$

In more detail, in both steps (b) and (c) the idea is to take as base functions the hard class from the previous step (respectively "monotone hard to learn to high accuracy," and "monotone hard to learn to moderate accuracy"), and compose them with a very noise-sensitive function in order to amplify hardness. Care must be taken to ensure that the combining function satisfies several necessary constraints (being monotone for (b) and k-alternating for (c), and being as sensitive as possible to the correct regime of noise in each case).

Useful tools

We begin by recalling a few notions and results that play a crucial role in our approach.

▶ **Definition 3.8** (Noise stability). For $f\colon \{0,1\}^n \to \{0,1\}$, the *noise stability* of f at $\eta \in [-1,1]$ is

$$\mathrm{Stab}_\eta(f) \stackrel{\text{def}}{=} 1 - 2\,\mathbf{Pr}[\, f(x) \neq f(y)\,]$$

where x is drawn uniformly at random from $\{0,1\}^n$ and y is obtained from x by independently for each bit having $\mathbf{Pr}[y_i = x_i] = (1+\eta)/2$ (i.e., x and y are η-*correlated*).

▶ **Definition 3.9** (Bias and expected bias). The *bias* of a Boolean function $h\colon \{0,1\}^n \to \{0,1\}$ is the quantity $\mathrm{bias}(h) \stackrel{\text{def}}{=} \max(\mathbf{Pr}[\, h = 1\,], \mathbf{Pr}[\, h = 0\,])$, while the *expected bias of h at δ* is defined as $\mathrm{ExpBias}_\delta(h) \stackrel{\text{def}}{=} \mathbf{E}_\rho[\mathrm{bias}(h_\rho)]$, where ρ is a random restriction on n coordinates where each coordinate is independently left free with probability δ and set to 0 or 1 with same probability $(1-\delta)/2$.

▶ **Fact 3.10** (Proposition 4.0.11 from [19]). *For $\delta \in [0,1/2]$ and $f\colon \{0,1\}^n \to \{0,1\}$, we have*

$$\frac{1}{2} + \frac{1}{2}\,\mathrm{Stab}_{1-2\delta}(f) \le \mathrm{ExpBias}_{2\delta}(f) \le \frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{Stab}_{1-2\delta}(f)}.$$

Building on Talagrand's probabilistic construction [28] of a class of functions that are sensitive to very small noise, Mossel and O'Donnell [18] gave the following noise stability upper bound. (We state below a slightly generalized version of their Theorem 3, which follows from their proof with some minor changes; see Appendix A.2 for details of these changes.)

▶ **Theorem 3.11** (Theorem 3 of [18]). *There exists an absolute constant K and an infinite family of* balanced *monotone functions $g_r\colon \{0,1\}^r \to \{0,1\}$ such that $\mathrm{Stab}_{1-\tau/\sqrt{r}}(g_r) \le 1 - K\tau$ holds for all sufficiently large r, as long as $\tau \in [16/\sqrt{r}, 1]$.*

Applying Fact 3.10, it follows that for the Mossel–O'Donnell function g_r on r inputs and any τ as above, we have

$$\frac{1}{2} \le \mathrm{ExpBias}_{\gamma}(g_r) \le \frac{1}{2} + \frac{1}{2}\sqrt{1-K\tau} \le 1 - \frac{K}{4}\tau \tag{2}$$

for $\gamma \stackrel{\text{def}}{=} \frac{\tau}{\sqrt{r}}$.

We will use the above upper bound on expected bias together with the following key tool from [7], which gives a hardness amplification result for uniform distribution learning. This result builds on the original hardness amplification ideas of O'Donnell [19]. (We note that the original theorem statement from [7] deals with the running time of learning algorithms, but inspection of the proof shows that the theorem also applies to the number of membership queries that the learning algorithms perform.)

▶ **Theorem 3.12** (Theorem 12 of [7]). *Fix $g\colon \{0,1\}^r \to \{0,1\}$, and let $\mathcal{F}$ be a class of m-variable Boolean functions such that for every $f \in \mathcal{F}$, $\mathrm{bias}(f) \le \frac{1}{2} + \frac{\epsilon}{8r}$. Let A be a uniform distribution membership query algorithm that learns $g \otimes \mathcal{F}$ to accuracy $\mathrm{ExpBias}_{\gamma}(g) + \epsilon$ using $T(m, r, 1/\epsilon, 1/\gamma)$ queries. Then there exists a uniform-distribution membership query algorithm B that learns $\mathcal{F}$ to accuracy $1-\gamma$ using $O(T \cdot \mathrm{poly}(m, r, 1/\epsilon, 1/\gamma))$ membership queries.*

Hardness of learning monotone functions to high accuracy. At the bottom level, corresponding to step (a) in (1), our approach relies on the following simple claim which states that monotone functions are hard to learn to very high accuracy. (We view this claim, as essentially folklore; as noted in the introduction it slightly sharpens a lower bound given in [6]. A proof is given for completeness in Appendix A.3.)

▶ **Claim 3.13** (A slice of hardness). *There exists a class of balanced monotone Boolean functions $\mathcal{G} = \{\mathcal{G}_m\}_{m\in\mathbb{N}}$ and a universal constant C such that, for any constants $0 < \alpha \le 1/10$, learning $\mathcal{G}_m$ to error $0 < \varepsilon \le \alpha/\sqrt{m}$ requires at least 2^{Cm} membership queries.*

We now prove Theorem 3.2, i.e. we establish a stronger lower bound (in terms of the range of accuracy it applies to) against learning the class of *monotone* functions. We do this by amplifying the hardness result of Fact 3.13 by composing the "mildly hard" class of functions $\mathcal{G}$ with a monotone function g – the Mossel–O'Donnell function of Theorem 3.11 – that is very sensitive to small noise (intuitively, the noise rate here is comparable to the error rate from Fact 3.13).

Proof of Theorem 3.2. We will show that there exists an absolute constant $\alpha > 0$ such that for any n sufficiently large and $\tau \in [\frac{1}{n^{1/6}}, 1/2 - c]$, there exist $m = m(n)$, $r = r(n)$ (both of which are $\omega_n(1)$) such that learning the class of (balanced) functions $\mathcal{H}_n = g_r \otimes \mathcal{G}_m$ on $n = mr$ variables to accuracy $1 - \tau$ requires at least $2^{\alpha\sqrt{n}/\tau}$ membership queries.

By contradiction, suppose we have an algorithm A which, for all m, r, τ as above, learns the class $\mathcal{H}_n$ to accuracy $1 - \tau$ using $T = T_A(n, \tau) < 2^{\alpha\sqrt{n}/\tau}$ membership queries. We show that this implies that for infinitely many values of m, one can learn $\mathcal{G}_m$ to error $\varepsilon = .1/\sqrt{m}$ with $2^{o(m)}$ membership queries, in contradiction to Fact 3.13.

Fix any n large enough and $\tau \in [\frac{1}{n^{1/6}}, .1]$, and choose m, r satisfying $mr = n$ and $\frac{5}{K} \cdot \frac{\tau}{\sqrt{r}} = \frac{.1}{\sqrt{m}}$, where K is the constant from Theorem 3.11. Note that this implies $m = \frac{K}{50} \cdot \frac{\sqrt{n}}{\tau} \in [\Theta(n^{1/2}), \Theta(n^{2/3})]$ so indeed both m and r are $\omega_n(1)$. (Intuitively, the value $\frac{.1}{\sqrt{m}}$ is the error we *want* to achieve to get a contradiction, while the value $\frac{5}{K} \cdot \frac{\tau}{\sqrt{r}}$ is the error we *can* get from Theorem 3.12.) Note that we indeed can use the Mossel–O'Donnell function from Theorem 3.11, which requires $\tau > \frac{16}{\sqrt{r}}$ – for our choice of r, this is equivalent to $\tau > \left(\frac{16\sqrt{K}}{\sqrt{50}}\right)^{2/3} \frac{1}{n^{1/6}}$. Finally, set $\varepsilon \stackrel{\text{def}}{=} .1/\sqrt{m}$.

We apply Theorem 3.12 with $g \stackrel{\text{def}}{=} g_r$, $\gamma = (5/K)\tau/\sqrt{r}$ and $\epsilon = \tau/4$. (Note that all functions in $\mathcal{G}_m$ are balanced, and thus trivially satisfy the condition that $\text{bias}(f) \leq \frac{\epsilon}{8r}$, and recall that $1 - \gamma$ is the accuracy the theorem guarantees against the original class $\mathcal{G}_m$.) With these parameters we have

$$\text{ExpBias}_\gamma(g) + \epsilon \underset{\text{Eq.(2)}}{\leq} 1 - \frac{K}{4}\frac{5\tau}{K} + \frac{\tau}{4} = 1 - \tau \leq \text{accuracy}(A).$$

Theorem 3.12 gives that there exists a learning algorithm B learning $\mathcal{G}_m$ to accuracy $1 - \gamma \geq 1 - \varepsilon$ with $T_B = O(T \cdot \text{poly}(m, r, 1/\tau, 1/\gamma)) = O(T \cdot \text{poly}(n, 1/\tau))$ membership queries, that is, $T_B = T_A(n, \tau) \cdot \text{poly}(n, 1/\tau) < 2^{\alpha\sqrt{n}/\tau + o(\sqrt{n}/\tau)}$ many queries. However, we have $2^{(\alpha+o(1))\sqrt{n}/\tau} = 2^{(\alpha+o(1))m \cdot \frac{\sqrt{n}}{\tau m}} < 2^{Cm}$, where the inequality comes from observing that $\frac{\sqrt{n}}{\tau m} = \frac{50}{K}$ (so that it suffices to pick α satisfying $50\alpha/K < C$). This contradicts Claim 3.13 and proves the theorem. ◀

▶ Remark (Improving this result). Proposition 1 of [18] gives a lower bound on the best noise stability that can be achieved by any monotone function. If this lower bound were in fact tight – that is, there exists a family of monotone functions $\{f_r\}$ such that for all $\gamma \in [-1, 1]$, $\text{Stab}_{1-\gamma}(f_r) = (1 - \gamma)^{(\sqrt{2/\pi} + o(1))\sqrt{r}}$ – then the above lower bound could be extended to an (almost) optimal range of τ, i.e. $\tau \in [\Phi(n)/\sqrt{n}, \frac{1}{2} - c]$ for Φ any fixed superconstant function.

From hardness of learning monotone Boolean functions to hardness of learning k-alternating functions. We now establish the hardness of learning k-alternating functions. Hereafter we denote by $\mathcal{H} = \{g_r \otimes \mathcal{G}_m\}_{m,r}$ the class of "hard" monotone functions from Theorem 3.2. Since g_r is balanced and every $f \in \mathcal{G}_m$ has bias $1/2$, it is easy to see that $\mathcal{H}$ is a class of balanced functions.

We begin by recalling the following useful fact about the noise stability of functions that are close to PAR:

▶ **Fact 3.14** (e.g., from the proof of Theorem 9 in [4]). *Let $r \geq 1$. If f is a Boolean function on r variables which η-approximates PAR_r, then for all $\delta \in [0, 1]$,*

$$\text{Stab}_{1-2\delta}(f) \leq (1 - 2\eta)^2(1 - 2\delta)^r + 4\eta(1 - \eta). \tag{3}$$

We use the above fact to define a function that is tailored to our needs: that is, a k-alternating function that is very sensitive to noise *and is defined on roughly k^2 inputs*. Without the last condition, one could just use PAR_k, but in our context this would only let us obtain a $\sqrt{k}$ (rather than a k) in the exponent of the lower bound, because of the loss in the reduction. To see why, observe that by using a combining function on k variables instead of k^2, the number of variables of the combined function $g_k \otimes \mathcal{G}_m$ would be only $n = km$. However, to get a contradiction with the hardness of monotone functions we shall need $k\sqrt{n}/\varepsilon \ll \sqrt{m}/\tau$, where $\tau \approx \varepsilon/k$, as the hardness amplification lemma requires the error to scale down with the number of combined functions.

▶ **Definition 3.15.** For any odd[2] $r \geq k \geq 1$, let $\mathsf{PAR}'_{k,r}$ be the symmetric Boolean function on r inputs defined as follows: for all $x \in \{0,1\}^r$,

$$\mathsf{PAR}'_{k,r}(x) = \begin{cases} 0 & \text{if } |x| \leq \frac{r-k}{2} \\ 1 & \text{if } |x| \geq \frac{r+k}{2} \\ \mathsf{PAR}_r(x) & \text{otherwise.} \end{cases}$$

In particular, $\mathsf{PAR}'_{k,r}$ is k-alternating, and agrees with PAR_r on the $k+1$ middle layers of the hypercube. By an additive Chernoff bound, one can show that $\mathsf{PAR}'_{k,r}$ is η-close to PAR_r, for $\eta = e^{-k^2/2r}$.

Proof of Theorem 3.4. $\mathcal{H}_n^{(k)}$ will be defined as the class $\mathsf{PAR}'_{k,r} \otimes \mathcal{H}_m$ for some r and m such that $n = mr$ (see below). It is easy to check that functions in $\mathcal{H}_n^{(k)}$ are balanced and k-alternating. We show below that for n sufficiently large, $2 \leq k < n^{1/14}$ and $\varepsilon \in [(1/300)(k^{14}/n)^{1/6}, \frac{1}{2} - c]$, learning $\mathcal{H}_n^{(k)}$ to accuracy $1-\varepsilon$ requires $2^{\Omega\left(k\sqrt{n}/\varepsilon\right)}$ membership queries.

By contradiction, suppose we have an algorithm A learning for all n, k, ε as above the class of k-alternating functions to accuracy $1-\varepsilon$ using $T_A(n,k,\varepsilon) < 2^{\beta\frac{k\sqrt{n}}{\varepsilon}}$ membership queries, where $\beta > 0$ is a universal constant to be determined during the analysis. We claim that this implies that for infinitely many values of m, one can learn $\mathcal{H}_m$ to some range of accuracies with a number of membership queries contradicting the lower bound of Theorem 3.2.

Fix any n large enough, k and ε as above (which in particular impose $k = O\left(n^{1/14}\right)$). The constraints we impose on m, r and τ are the following:

$$mr = n; \quad \mathrm{ExpBias}_\tau(\mathsf{PAR}'_{k,r}) + \varepsilon \leq 1 - \varepsilon; \quad m = \omega_n(1); \quad \tau \geq \frac{1}{m^{1/6}}; \tag{4}$$

$$\beta k \frac{\sqrt{n}}{\varepsilon} < \alpha \frac{\sqrt{m}}{\tau}, \tag{5}$$

where the constraints in (4) are for us to apply the previous theorems and lemmas, while (5) is needed to ultimately derive a contradiction.

One can show that by taking $r \stackrel{\text{def}}{=} \left\lfloor \frac{k^2}{2\ln 5} \right\rfloor \geq 1$ and $\tau \stackrel{\text{def}}{=} \frac{100\varepsilon}{r}$, the second constraint of (4) is satisfied, as then $\mathrm{Stab}_{1-\tau}(\mathsf{PAR}'_{k,r}) \leq 1 - 8\varepsilon$ (for the derivation, see Appendix A.4). Then, with the first constraint of (4), we get (omitting for simplicity the floors) $m \stackrel{\text{def}}{=} \frac{n\tau}{100\varepsilon} = (2\ln 5)\frac{n}{k^2}$,

[2] The above definition can be straightforwardly extended to $r \geq k \geq 1$ not necessarily odd, resulting in a similar k-alternating perfectly balanced function $\mathsf{PAR}'_{k,r}$ that agrees with PAR_r on $k + O(1)$ middle layers of the cube and is 0 below and 1 above those layers. For the sake of simplicity we leave out the detailed description of the other cases.

so as long as $k = o(\sqrt{n})$, the third constraint of (4) is met as well. With these settings, the final constraint of (4) can be rewritten as $\varepsilon \geq \frac{1}{100}\left(\frac{r^7}{n}\right)^{1/6} = \frac{1}{100(2\ln 5)^{7/6}}\left(\frac{k^{14}}{n}\right)^{1/6}$. As $(2\ln 5)^{7/6} > 3$, it is sufficient to have $\varepsilon \geq \frac{1}{300}\left(\frac{k^{14}}{n}\right)^{1/6}$, which holds because of the lower bound on ε.

It only remains to check Constraint (5) holds:

$$k\frac{\sqrt{n}}{\varepsilon} = 100k\frac{\sqrt{n}}{\tau r} = 100\frac{k}{\sqrt{r}}\frac{\sqrt{m}}{\tau} \leq \left(100\sqrt{\frac{2\ln 5}{1-2\ln 5/k^2}}\right)\frac{\sqrt{m}}{\tau} \leq 300\sqrt{2\ln 5}\cdot\frac{\sqrt{m}}{\tau},$$

where the first inequality holds because as $\frac{1}{r} \leq \frac{1}{\frac{k^2}{2\ln 5}-1}$ and the second holds because $k \geq 2$. So for the right choice of $\beta = \Omega(1)$, e.g. $\beta = \alpha/600$, $\beta k\frac{\sqrt{n}}{\varepsilon} < \alpha\frac{\sqrt{m}}{\tau}$, and (5) is satisfied.

It now suffices to apply Theorem 3.12 to $\mathsf{PAR}'_{k,r} \otimes \mathcal{H}_m$, with parameters $\gamma = \tau$ and ε, on algorithm A, which has accuracy $\mathrm{acc}(A) \geq 1-\tau \geq \mathrm{ExpBias}_\gamma(\mathsf{PAR}'_{k,r}) + \epsilon$. Since the functions of $\mathcal{H}$ are unbiased, it follows that there exists an algorithm B learning $\mathcal{H}_m$ to accuracy $1-\tau$, with $\tau > 1/2m^{1/6}$, making only

$$T_B(m,\tau) = O(T_A(n,k,\varepsilon)\,\mathrm{poly}(n,k,1/\varepsilon)) = 2^{\beta k\frac{\sqrt{n}}{\varepsilon}(1+o(1))} < 2^{\alpha\frac{\sqrt{m}}{\tau}}$$

membership queries, which contradicts the lower bound of Theorem 3.2. ◀

▶ **Observation 3.16** (On the relation between ε and k)**.** *The tradeoff in the ranges for k and ε appear to be inherent to this approach. Namely, it comes essentially from Constraint* (4)*, itself deriving from the hypotheses of Theorem 3.2. However, even getting an optimal range in the latter would still require $\tau = \Omega(1/\sqrt{m})$, which along with $r \approx k^2$ and $\tau \approx \varepsilon/r$ impose $k = O(n^{1/6})$ and $\varepsilon = \Omega(k^3/\sqrt{n})$.*

References

1. K. Amano and A. Maruoka. On learning monotone Boolean functions under the uniform distribution. In *International Conference on Algorithmic Learning Theory* (ALT), pages 57–68, 2002.
2. K. Amano and A. Maruoka. A Superpolynomial Lower Bound for a Circuit Computing the Clique Function with At Most $(1/6)\log\log n$ Negation Gates. *SIAM Journal on Computing*, 35(1):201–216, 2005.
3. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
4. E. Blais and L-Y. Tan. Approximating Boolean functions with depth-2 circuits. In *Conference on Computational Complexity* (CCC), pages 74–85, 2013.
5. A. Blum, C. Burch, and J. Langford. On learning monotone Boolean functions. In *Symposium on Foundations of Computer Science* (FOCS), pages 408–415, 1998.
6. N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.
7. V. Feldman, H. K. Lee, and R. A. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. *Journal of Machine Learning Research – Proceedings Track*, 19:273–292, 2011.
8. O. Goldreich and R. Izsak. Monotone circuits: One-way functions versus pseudorandom generators. *Theory of Computing*, 8(1):231–238, 2012.
9. S. Guo and I. Komargodski. Negation-limited formulas. Technical Report 22(26), Electronic Colloquium on Computational Complexity (ECCC), 2015.

10 S. Guo, T. Malkin, I. C. Oliveira, and A. Rosen. The power of negations in cryptography. In *Theory of Cryptography Conference* (TCC), pages 36–65, 2015.

11 M. Kearns and L. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

12 A. D. Korshunov. Monotone Boolean functions. *Russian Mathematical Surveys* (*Uspekhi Matematicheskikh Nauk*), 58(5):929–1001, 2003.

13 N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

14 A. A. Markov. On the inversion complexity of systems of functions. *Doklady Akademii Nauk SSSR*, 116:917–919, 1957. English translation in [15].

15 A. A. Markov. On the inversion complexity of a system of functions. *Journal of the ACM*, 5(4):331–334, October 1958.

16 H. Morizumi. Limiting negations in formulas. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 701–712, 2009.

17 H. Morizumi. Limiting negations in non-deterministic circuits. *Theoretical Computer Science*, 410(38-40):3988–3994, 2009.

18 E. Mossel and R. O'Donnell. On the noise sensitivity of monotone functions. *Random Structures and Algorithms*, 23(3):333–350, 2003.

19 R. O'Donnell. *Computational applications of noise sensitivity*. PhD thesis, MIT, June 2003.

20 R. O'Donnell and R. Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007.

21 R. O'Donnell and K. Wimmer. KKL, Kruskal-Katona, and monotone nets. *SIAM Journal on Computing*, 42(6):2375–2399, 2013.

22 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992.

23 A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Doklady Akademii Nauk SSSR*, 281:798–801, 1985. English translation in: Soviet Mathematics Doklady 31:354–357, 1985.

24 B. Rossman. Correlation bounds against monotone NC^1. In *Conference on Computational Complexity* (CCC), 2015.

25 M. Santha and C. Wilson. Limiting negations in constant depth circuits. *SIAM Journal on Computing*, 22(2):294–302, 1993.

26 R. Servedio. On learning monotone DNF under product distributions. *Information and Computation*, 193(1):57–74, 2004.

27 S. Sung and K. Tanaka. Limiting Negations in Bounded-Depth Circuits: an Extension of Markov's Theorem. In *International Symposium on Algorithms and Computation* (ISAAC), pages 108–116, 2003.

28 M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.

A Proofs

A.1 Proof of Claim 2.2

Suppose $\mathbf{Inf}[f] \geq \alpha n$ for some $\alpha \in (0,1]$: this means that at least an α fraction of all edges are bichromatic. Define the *weight level* k (denoted $\mathcal{W}_k$) to be the set of all edges going from a vertex of Hamming weight k to a vertex of Hamming weight $k+1$ (in particular, $|\mathcal{W}_k| = (n-k)\binom{n}{k}$), and consider weight levels $n/2 - a\sqrt{n}, \ldots, n/2 + a\sqrt{n}-1$ (the "middle levels") for $a \stackrel{\text{def}}{=} \sqrt{(1/2)\ln(8/\alpha)}$. (We suppose without loss of generality that $n/2 - a\sqrt{n}$ is

a whole number.) Now, the fraction of all edges which do *not* lie in these middle levels is at most

$$\frac{1}{n2^{n-1}} \cdot 2 \sum_{j=0}^{\frac{n}{2}-a\sqrt{n}-1} |\mathcal{W}_k| \leq \frac{2n}{n2^{n-1}} \sum_{j=0}^{\frac{n}{2}-a\sqrt{n}-1} \binom{n}{k} \leq \frac{4}{2^n} \sum_{j=1}^{\frac{n}{2}-a\sqrt{n}-1} \binom{n}{k} \leq 4e^{-2a^2} = \frac{\alpha}{2}.$$

So no matter how many of these edges are bichromatic, it must still be the case that at least an $\alpha/2$ fraction of all edges in the "middle levels" are bichromatic.

Since the ratio

$$\frac{|\mathcal{W}_{n/2}|}{|\mathcal{W}_{n/2-a\sqrt{n}}|} = \frac{\frac{n}{2}\binom{n}{n/2}}{\left(\frac{n}{2}+a\sqrt{n}\right)\binom{n}{n/2-a\sqrt{n}}}$$

converges monotonically from below (when n goes to infinity) to $C \stackrel{\text{def}}{=} e^{2a^2}$, any two weight levels amongst the middle ones have roughly the same number of edges, up to a multiplicative factor C. Setting $p = \alpha/6C$ and $q = \alpha/6$, this implies that at least a p fraction of the weight levels in the middle levels have at least a q fraction of their edges being bichromatic. (Indeed, otherwise we would have, letting b_k denote the number of bichromatic edges in weight layer k,

$$\begin{aligned}
\frac{\alpha}{2} \cdot \underbrace{\sum_{k=\frac{n}{2}-a\sqrt{n}}^{\frac{n}{2}+a\sqrt{n}-1} |\mathcal{W}_k|}_{\text{total}} &\leq \sum_{k=\frac{n}{2}-a\sqrt{n}}^{\frac{n}{2}+a\sqrt{n}-1} b_k \\
&\leq \sum_{\substack{k \in [\frac{n}{2}-a\sqrt{n}, \frac{n}{2}+a\sqrt{n}-1] \\ b_k > q|\mathcal{W}_k|}} |\mathcal{W}_k| + \sum_{\substack{k \in [\frac{n}{2}a\sqrt{n}, \frac{n}{2}+a\sqrt{n}-1] \\ b_k \leq q|\mathcal{W}_k|}} q \cdot |\mathcal{W}_k| \\
&\leq p \cdot 2a\sqrt{n} \cdot \left|\mathcal{W}_{n/2}\right| + q \cdot \sum_{k=\frac{n}{2}-a\sqrt{n}}^{\frac{n}{2}+a\sqrt{n}-1} |\mathcal{W}_k| \\
&\leq p \cdot C \cdot \sum_{k=\frac{n}{2}-a\sqrt{n}}^{\frac{n}{2}+a\sqrt{n}-1} |\mathcal{W}_k| + q \cdot \sum_{k=\frac{n}{2}-a\sqrt{n}}^{\frac{n}{2}+a\sqrt{n}-1} |\mathcal{W}_k| .
\end{aligned}$$

So $\frac{\alpha}{2} \cdot \text{total} \leq p \cdot C \cdot \text{total} + q \cdot .\text{total}$, which gives $\frac{\alpha}{2} \leq \frac{\alpha}{6C} \cdot C + \frac{\alpha}{6} = \frac{\alpha}{3}$, a contradiction.)

Let S be this collection of at least $2a\sqrt{n}p$ weight levels (from the middle ones) that each have at least a q fraction of edges being bichromatic, and write p_i to denote the fraction of bichromatic edges in $\mathcal{W}_i$, so that for each $i \in S$ it holds that $p_i \geq q$. Consider a random chain from 0^n to 1^n. The marginal distribution according to which an edge is drawn from any given fixed weight level i is uniform on $\mathcal{W}_i$, so by linearity, the expected number of bichromatic edges in a random chain is at least $\sum_{i \in S} p_i \geq 2a\sqrt{n}pq = \Omega(\sqrt{n})$, and hence some chain must have that many bichromatic edges. ◀

A.2 Derivation of Theorem 3.11 using Theorem 3 of [18]

The original theorem is stated for $\tau = 1$, with the upper bound being $1 - \Omega(1)$. However, the proof of [18] goes through for our purposes until the very end, where they set $\epsilon \stackrel{\text{def}}{=} \frac{1}{\sqrt{r}}$ and need to show that

$$e^{-2}\left(1 - (1 - \epsilon + 2\sqrt{\epsilon/r})^{\sqrt{r}}\right) = \Omega(1).$$

More precisely, the proof goes overall as follows: for some realization of the Talagrand function on r variables g_r, we want (for some absolute constant K) that

$$1 - K\tau \geq \mathrm{Stab}_{1-\frac{\tau}{\sqrt{r}}}(g_r) = 1 - 2\,\mathbf{Pr}\Big[\, g_r \circ N_{1-\frac{\tau}{\sqrt{r}}}(x) \neq g_r(x) \,\Big].$$

That is, one needs to show $\mathbf{Pr}\Big[\, g_r \circ N_{1-\frac{\tau}{\sqrt{r}}}(x) \neq g_r(x) \,\Big] \geq \frac{K}{2}\tau$; and in turn, it is sufficient to prove that for g a random Talagrand function on r variables,

$$\mathbf{E}_g\Big[\, \mathbf{Pr}\Big[\, g \circ N_{1-\frac{\tau}{\sqrt{r}}}(x) \neq g(x) \,\Big] \,\Big] \geq \frac{K}{2}\tau.$$

This is where we slightly adapt the [18] proof. Where they set a parameter ϵ to be equal to $1/\sqrt{r}$ and analyze $\mathbf{E}_g[\mathbf{Pr}[\, g \circ N_{1-2\epsilon}(x) \neq g(x)\,]]$, we set for our purposes $\epsilon \stackrel{\text{def}}{=} \frac{\tau}{2\sqrt{r}}$. The rest of the argument goes through until the very end, where it only remains to show that

$$ae^{-2}\left(1 - (1 - \epsilon + 2\sqrt{\epsilon/r})^{\sqrt{r}}\right) \geq \frac{K}{2}\tau \tag{6}$$

(a being a small constant resulting from the various conditionings in their proof), or equivalently, that $(1 - \epsilon + 2\sqrt{\epsilon/r})^{\sqrt{r}} \leq 1 - \frac{e^2K}{2a}\tau$. But the left-hand side can be rewritten as

$$\begin{aligned}
(1 - \epsilon + 2\sqrt{\epsilon/r})^{\sqrt{r}} &= e^{\sqrt{r}\ln(1-\epsilon+2\sqrt{\epsilon/r})} = e^{\sqrt{r}\ln(1-\tau/2\sqrt{r}+\sqrt{2\tau}/r^{3/4})} \\
&= e^{\sqrt{r}\ln\left(1-\frac{\tau}{2\sqrt{r}}\left(1-\frac{2\sqrt{2}}{\sqrt{r^{1/2}\tau}}\right)\right)} \\
&\leq e^{-\sqrt{r}\cdot\frac{\tau}{2\sqrt{r}}\left(1-\frac{2\sqrt{2}}{\sqrt{r^{1/2}\tau}}\right)} && \left(\text{as } \tfrac{\tau}{2\sqrt{r}}\left(1-\tfrac{2\sqrt{2}}{\sqrt{r^{1/2}\tau}}\right) < 1\right) \\
&= e^{-\frac{\tau}{2}\left(1-\frac{2\sqrt{2}}{\sqrt{r^{1/2}\tau}}\right)} \leq e^{-\frac{\tau}{2}(1-\frac{1}{\sqrt{2}})} && \left(\text{as } \tau > \tfrac{16}{\sqrt{r}}\right) \\
&\leq e^{-\frac{\tau}{7}} \leq 1 - \frac{\tau}{8} \leq 1 - \frac{e^2K}{2a}\tau. \\
& && \text{(first as } \tau < 1\text{, then for a suitable choice of } K\text{)}
\end{aligned}$$

◀

A.3 Proof of Fact 3.13

We give the proof for m even; by standard techniques, it extends easily to the odd case. For any $m \in 2\mathbb{N}$, define $\mathcal{C}_m$ as the class of functions f generated as follows: let $R = \{\, x \in \{0,1\}^m : \ |x| = m/2 \,\}$, and partition R in $|R|/2$ pairs of elements $(x^\ell, \bar{x}^\ell)$. For all $x \in \{0,1\}^m$,

$$f(x) = \begin{cases} 0 & \text{if } |x| < m/2 \\ r_\ell & \text{if } x \in R \text{ and } x = x^\ell \\ 1 - r_\ell & \text{if } x \in R \text{ and } x = \bar{x}^\ell \\ 1 & \text{if } |x| > m/2 \end{cases}$$

where the $|R|/2$ bits r_ℓ are chosen independently and uniformly at random. Clearly, f is balanced, and we have

$$|R| = \binom{m}{m/2} \underset{m\to\infty}{\sim} \sqrt{\frac{2}{\pi}} \cdot \frac{2^m}{\sqrt{m}} \stackrel{\text{def}}{=} \gamma 2^m.$$

Suppose we have a learning algorithm A for $\mathcal{C}_m$ making $q < 2^{Cm}$ membership queries. Fix $0 < \alpha \leq 1$, and $\varepsilon = \alpha/\sqrt{m}$; to achieve error at most ε overall, A must in particular achieve error at most $\frac{\varepsilon}{\gamma} = \sqrt{\frac{\pi}{2}}\alpha$ on R. But after making q queries, there are still at least $t = \gamma 2^m/2 - 2^{Cm} > 0.99\,|R|$ points in R (for m big enough) A has not queried, and hence with values chosen uniformly at random; on each of these points, A is wrong with probability exactly half, and in particular

$$\begin{aligned}\mathbf{Pr}\left[\text{error } \leq \frac{\varepsilon}{\gamma}\right] < \mathbf{Pr}[\text{error } \leq 2\alpha] &= \mathbf{Pr}\left[\sum_{i=1}^{t} X_i \leq 2\alpha\,|R|\right] \\ &\leq \mathbf{Pr}\left[\sum_{i=1}^{t} X_i \leq \frac{200}{99}\alpha t\right] \\ &\leq e^{-\frac{(1-\frac{400}{99}\alpha)^2 t}{2}} = o(1)\end{aligned}$$

with an additive Chernoff bound. This means that with high probability over the choice of the target concept, A will fail to learn it to accuracy $1 - \varepsilon$. ◀

A.4 Derivation of the bound $\mathrm{Stab}_{1-\tau}(\mathsf{PAR}'_{k,r}) \leq 1 - 8\varepsilon$

By setting r as stated we get that $r \leq k^2/\ln(1/\varepsilon)$ and the distance between $\mathsf{PAR}'_{k,r}$ and PAR_r becomes $\eta = e^{-k^2/2r} \leq 1/5$. Since we aim at having $\mathrm{ExpBias}_\tau(\mathsf{PAR}'_{k,r}) \leq 1 - 2\varepsilon$, it is sufficient to have $\sqrt{\mathrm{Stab}_{1-\tau}(\mathsf{PAR}'_{k,r})} \leq 1 - 4\varepsilon$; which would in turn be implied by $\mathrm{Stab}_{1-\tau}(\mathsf{PAR}'_{k,r}) \leq 1 - 8\varepsilon$.

By Fact 3.14, it is sufficient to show that $(1-2\eta)^2(1-\tau)^r + 4\eta(1-\eta) \leq 1 - 8\varepsilon$; note that since $\varepsilon < 1/100$ and by our choice of τ,

$$\begin{aligned}(1-2\eta)^2(1-\tau)^r + 4\eta(1-\eta) \leq \frac{(1-2\eta)^2}{1+100\varepsilon} + 4\eta(1-\eta) &\leq (1-2\eta)^2(1-50\varepsilon) + 4\eta(1-\eta) \\ &\leq (1-4\eta+4\eta^2)(1-50\varepsilon) + 4\eta(1-\eta) \\ &= 1 - 4\eta - 50\varepsilon + 200\eta\varepsilon + 4\eta^2 - 200\varepsilon\eta^2 + 4\eta - 4\eta^2 \\ &= 1 - 50\varepsilon + 200\varepsilon\eta(1-\eta) \leq 1 - 50\varepsilon + 32\varepsilon = 1 - 18\varepsilon \\ &\leq 1 - 8\varepsilon.\end{aligned}$$

◀