On the noise sensitivity of monotone functions

Elchanan Mossel, Ryan O'Donnell

ABSTRACT:

It is known that for all monotone functions $f : \{0,1\}^n \to \{0,1\}$, if $x \in \{0,1\}^n$ is chosen uniformly at random and y is obtained from x by flipping each of the bits of x independently with probability ϵ, then $\mathbf{P}[f_n(x) \neq f_n(y)] < c\epsilon\sqrt{n}$, for some $c > 0$.

Previously, the best construction of monotone functions satisfying $\mathbf{P}[f_n(x) \neq f_n(y)] \geq \delta$, where $0 < \delta < 1/2$, required $\epsilon \geq c(\delta)n^{-\alpha}$, where $\alpha = 1 - \ln 2/\ln 3 = 0.36907\ldots$, and $c(\delta) > 0$. We improve this result by achieving for every $0 < \delta < 1/2$, $\mathbf{P}[f_n(x) \neq f_n(y)] \geq \delta$, with:

- *$\epsilon = c(\delta)n^{-\alpha}$ for any $\alpha < 1/2$, using the recursive majority function with arity $k = k(\alpha)$;*
- *$\epsilon = c(\delta)n^{-1/2}\log^t n$ for $t = \log_2\sqrt{\pi/2} = .3257\ldots$, using an explicit recursive majority function with increasing arities; and,*
- *$\epsilon = c(\delta)n^{-1/2}$, non-constructively, following a probabilistic CNF construction due to Talagrand.*

The constructions have implications for learning theory, computational complexity, and neural networks, and they shed some light on the American electoral system.

1 Introduction

1.1 Noise sensitivity and Fourier coefficients

The papers [KKL88, BL90] suggested the importance of the *Fourier expansion* and the *influence of variables* on f for the study of boolean functions. The ideas developed in these papers proved to be extremely fruitful in later work, e.g., [LMN93, FK96, F98, BKS98] and the material in Subsection 2, to name just a few examples.

Let $\Omega_n = \{-1,+1\}^n$ be the Hamming cube endowed with the uniform probability measure $\mathbf{P}$. We look at boolean functions $f : \Omega_n \to \{-1,+1\}$. We are mostly concerned with *monotone* boolean functions. Recall that a function f is *monotone* if for all $x, y \in \Omega_n$ we have $f(x) \leq f(y)$ whenever $x \leq y$ (in the sense $x_i \leq y_i$ for all i).

For $-1 \leq \eta \leq 1$ and $x \in \Omega_n$, define $N_\eta(x)$ to be a random element y of Ω_n which satisfies $\mathbf{E}[y_i x_i] = \eta$ (equivalently, $\mathbf{P}[x_i \neq y_i] = (1-\eta)/2$) independently for all i. It is natural to measure how stable f is to η-noise by the correlation between $f(x)$ and $f(N_\eta(x))$,

$$Z(f,\eta) = \mathbf{E}[f(N_\eta(x))f(x)] = 1 - 2\mathbf{P}[f(N_\eta(x)) \neq f(x)]. \tag{1}$$

If f is stable under the noise operator N_η, then typically $f(x)$ and $f(N_\eta(x))$ should have the same value and therefore $Z(f,\eta)$, the expression in (1), should be close to 1; if f is sensitive to noise, then $Z(f,\eta)$ should be close to 0.

The space Ω_n with the uniform probability measure naturally gives rise to an inner product space on all functions $f : \Omega_n \to \mathbb{R}$:

$$\langle f, g\rangle = \mathbf{E}[fg] = 2^{-n} \sum_{x\in\Omega_n} f(x)g(x).$$

For a set $S \subseteq [n]$, define $u_S(x) = \prod_{i\in S} x_i$. Since $u_S u_{S'} = u_{S\Delta S'}$, where Δ denotes symmetric difference, it follows that $(u_S)_{S\subseteq[n]}$ is an orthonormal basis. We call $\hat{f}(S) = \langle u_S, f\rangle$ the S *Fourier coefficient* of f, and $f = \sum_{S\subseteq[n]} \hat{f}(S)u_S$ the *Fourier expansion* of f.

The basis $(u_S)_{S\subseteq[n]}$ has very nice properties with respect to the noise operator; most notably, for all x and S, $\mathbf{E}[u_S(N_\eta(x))] = \eta^{|S|}u_S(x)$, which implies

$$Z(f, \eta) = \mathbf{E}[f(N_\eta(x))f(x)] = \sum_{S\subseteq[n]} \eta^{|S|}\hat{f}^2(S) \tag{2}$$

(see e.g. [BKS98, BJT99, O02]).

The stability of the function f under noise, $Z(f, \eta)$, is therefore closely related to how much of the ℓ_2 mass of the Fourier coefficients of f lies on coefficients $\hat{f}(S)$ for large sets S.

In addition to the sum in (2), it is common to study several other weighted sums of f's squared Fourier coefficients. By Parseval's identity, $\sum_S \hat{f}^2(S) = 1$. The average sensitivity of f is defined by $I(f) := \sum_S |S|\hat{f}^2(S)$. It is shown in [KKL88] that $I(f) = \sum_{k=1}^n I_k(f)$, where $I_k(f)$ is the probability the value of the function flips, when the k'th bit is flipped. Note that if f is monotone, then $I_k(f) = |\hat{f}(\{k\})|$. Finally, we have the quantity $II(f) := \sum_{k=1}^n I_k^2(f)$, introduced in [BKS98].

1.2 Sensitivity of monotone functions

The parity function, $f = u_{[n]} = \oplus$, is the boolean function most sensitive to noise: $Z(f, \eta) = \eta^n$ is minimal, and $I(f) = n$ is maximal.

It is natural to ask if monotone functions can be as sensitive to noise as non-monotone functions. It is known (see Lemma 6.1 of [FK96]) that the majority function has maximal I among all monotone functions on n inputs. Since its average sensitivity is easily computed to be $\sqrt{2/\pi}\sqrt{n} + o(\sqrt{n})$, we get that for all for all monotone f on n inputs,

$$I(f) \le (\sqrt{2/\pi} + o(1))\sqrt{n}. \tag{3}$$

It remains to determine how small $N_\eta(f)$ can be for monotone functions. A natural goal is to find a monotone function f on n bits such that $Z(f, 1-\delta) \le 1 - \Omega(1)$ for the smallest possible quantity δ. This problem was implicitly posed in [BKS98].

An easy folklore argument (see long version for proof) uses (3) to deduce:

Proposition 1.1 *For all monotone f on n inputs,*

$$Z(f, 1-\delta) \ge (1-\delta)^{(1+o(1))\sqrt{(2/\pi)n}}.$$

Therefore if $Z(f, 1-\delta) \le 1-\epsilon$, then $\delta \ge \sqrt{\frac{2}{\pi}}\frac{\epsilon}{\sqrt{n}} + o(1/\sqrt{n})$.

In particular, in order to obtain $Z(f, 1-\delta) \leq 1-\Omega(1)$, δ must satisfy $\delta \geq \Omega(n^{-1/2})$. Prior to this work, the best sensitivity with respect to N_η was achieved via the recursive majority of 3 function (folklore, see [BL90, BKS98]). This function satisfies $Z(f, 1-\delta) \leq 1-\Omega(1)$, for $\delta = n^{-\alpha}$, where $\alpha = 1 - \ln 2/\ln 3 = 0.36907\ldots$

1.3 Our results

Recursive majority functions seem to be sensitive to noise. Previous techniques for analyzing recursive majorities had suggested that recursive majority of 5, 7, etc. might be less sensitive than recursive majority of 3. However, this is not the case.

Theorem 1.2 *Let $k = 2r+1$ and let $\mathsf{REC\text{-}MAJ\text{-}}k_\ell$ denote the ℓ level k recursive majority. Let*

$$b = \frac{2r+1}{2^{4r}}\binom{2r}{r}^2, \qquad a = \frac{2r+1}{2^{2r}}\binom{2r}{r}.$$

Then $Z(\mathsf{REC\text{-}MAJ\text{-}}k_\ell, 1-\delta) \leq \epsilon$ for $\ell \geq \Big(\log_a(1/\delta) + \log_{1/b}(1/\epsilon)\Big)(1 + r(\epsilon, \delta))$, where $r(\epsilon, \delta) \to 0$ as $\epsilon \to 0$ and $\delta \to 0$. Hence for every $\alpha < 1/2$, and $0 < \delta < 1$, there exists an odd $k \geq 3$ such that for $n = t^\ell$, $f_n = \mathsf{REC\text{-}MAJ\text{-}}k_\ell : \Omega_n \to \{-1, +1\}$ is a balanced function with

$$Z(f_n, 1 - n^{-\alpha}) \leq 1 - \delta + o(1).$$

Note that this construction is explicit. Moreover, using k-majority gates, we obtain a read-once, log-depth circuit which implements the function. The proof technique is closely related to techniques in classical branching processes [AN72] (see also [M98]).

By relaxing the bounded degree property, and using instead majority gates of varying fan-in, we obtain an explicit read-once construction of log log-depth which is sensitive to a noise rate of about $n^{-1/2}$, up to a sub-logarithmic correction.

Theorem 1.3 *For every $0 < \delta < 1$, there exists an explicit infinite family of balanced monotone functions $f_n : \Omega_n \to \{-1, +1\}$ with the following property:*

$$Z(f_n, 1 - 1/M) \leq 1 - \delta + o(1),$$

where $M = \sqrt{n}/\Theta(\log^t n)$, and $t = \log_2 \sqrt{\pi/2} = .3257\ldots$.

It is interesting to note that the t parameter is optimal for this construction.

Finally, analyzing a probabilistic construction due to Talagrand [T96], we obtain a tight result up to constant factors.

Theorem 1.4 *For every $0 < \delta < 1$, there exists an infinite family of monotone functions $f_n : \Omega_n \to \{-1, +1\}$ with the following property:*

$$Z(f_n, 1 - n^{-1/2}) \leq 1 - \delta + o(1). \tag{4}$$

In this extended abstract, we sketch the proof of a slightly weakened version of Theorem 1.4, i.e., instead of (4) we prove

$$Z(f_n, 1 - n^{-1/2}) \leq 1 - \Omega(1). \tag{5}$$

2 Implications for other problems

2.1 Learning monotone functions

In the field of computational learning theory, one of the most widely studied models is Valiant's Probably Approximately Correct (PAC) model [V84]. In PAC learning, a *concept class* $\mathcal{C}$ is a collection $\cup_{n\geq 1}\mathcal{C}_n$ of boolean functions, where each function (concept) $f \in \mathcal{C}_n$ is a boolean function on n bits. Let $f \in \mathcal{C}_n$ be an unknown *target function*, and let $\mathcal{D}$ be an unknown probability *distribution* on $\{-1,+1\}^n$. A learning algorithm A for $\mathcal{C}$ takes as input an *accuracy parameter* $0 < \epsilon < 1$ and a *confidence parameter* $0 < \delta < 1$. During its execution, A has access to an *example oracle* $\mathsf{EX}(f)$ which, when queried, generates a random labeled example $\langle x, f(x)\rangle$, where x is drawn from distribution $\mathcal{D}$. A's goal is to output a hypothesis h which is a boolean function on n bits, which is "close" to f under distribution $\mathcal{D}$. Specifically, we say that A is a *PAC learning algorithm for* $\mathcal{C}$ if for every $f \in C$ and every ϵ, δ, with probability $1-\delta$ algorithm A outputs a hypothesis h satisfying $\Pr_{x\leftarrow\mathcal{D}}[f(x) \neq h(x)] \leq \epsilon$. Ideally one likes for A to run in time $\mathrm{poly}(n, s, 1/\epsilon, \log(1/\delta))$, where s is a "size parameter" of the concept class.

An important and well-studied restriction of the PAC model is *uniform PAC learning*, which is simply the case in which $\mathcal{D}$ is the uniform distribution on $\{-1,+1\}^n$. Linial, Mansour, and Nisan [LMN93] introduced a very powerful and general uniform PAC learning algorithm, which has come to be known as the "low degree algorithm" (see Mansour's survey [M94]). The low degree algorithm works for any concept class which has a *Fourier concentration bound*. Specifically, suppose that for every function f in a given concept class, $\sum_{|S|\geq m} \hat{f}^2(S) \leq \epsilon$. Then the low degree algorithm will PAC-learn this class under the uniform distribution in time $\exp(O(m\log(n/m)))\log(1/\delta)$. The algorithm works by drawing many examples for f, and using these to calculate empirical estimates for all Fourier coefficients $\hat{f}(S)$ with $|S| < m$. The hypothesis outputted is simply the sign of the resulting truncated Fourier expansion.

Bshouty and Tamon [BT96] give the fastest known uniform PAC learning algorithm for the concept class of monotone functions. Their algorithm is the low degree algorithm, and they show a Fourier concentration bound for the class of monotone functions with $m = O(\epsilon^{-1}\sqrt{n})$. (It is simple to derive this from (3); Bshouty and Tamon also extend these results to general product distributions on $\{-1,+1\}^n$.) This leads to a learning algorithm running in time $\exp(O(\frac{1}{\epsilon}\sqrt{n}\log(\epsilon\sqrt{n})))\log(1/\delta)$.

As a tightness result, [BT96] prove via a counting argument that there is a monotone f which does not satisfy $\sum_{|S|\geq m} \hat{f}^2(S) \leq n^{-1/2}\log n$ unless $m = \Omega(n)$. However this leaves open the question of $\epsilon = \Omega(n^{-1/2}\log n)$. To show that the low degree algorithm for monotone functions cannot be improved, we need to exhibit a monotone f for which $\sum_{|S|\geq\Omega(\epsilon^{-1}\sqrt{n})} \hat{f}^2(S) > \epsilon$. The functions f from Theorem 1.4 satisfy $\sum_{|S|\geq\sqrt{n}} \hat{f}^2(S) \geq \Omega(1)$. Hence the low degree algorithm will have $\Omega(1)$ error unless it goes up to degree $\sqrt{n}$. In fact, our Corollary 7.2 gives us an explicit function f with $\sum_{|S|\geq\tilde{\Omega}(\sqrt{\epsilon n})} \hat{f}^2(S) \geq 1-\epsilon$.

See [BJT99, KOS02] for more on noise sensitivity in the context of computational learning theory.

2.2 Hardness amplification within NP

The central problem in computational complexity theory is whether or not $\mathsf{NP} = \mathsf{P}$; i.e., deciding if proving a proposition is harder than verifying the proof of that proposition. In studying this problem, many researchers have considered the slightly weaker question of whether or not every language in NP can be computed by circuits of polynomial size. (See any standard text such as [Pa93, BDG88, DK00] for the definitions of P, NP, circuits, etc.) Let us phrase this question precisely. A language $F \in \mathsf{NP}$ gives rise to a family of characteristic functions $\langle f_n \rangle$, where $f_n : \{0,1\}^n \to \{0,1\}$ is defined by $f_n(x) = 1$ iff $x \in F$. We often abuse language by saying f_n is a function NP (we always have a particular family of functions in mind). A family of boolean circuits $\langle C_n \rangle$ is said to have polynomial size if there is a finite k such that $\text{size}(C_n) \leq O(n^k)$. We say NP has polynomial-sized circuits if for every family of functions $\langle f_n \rangle$ in NP, there is a circuit family $\langle C_n \rangle$ of polynomial size such that $C_{|x|}(x) = f_{|x|}(x)$ for every boolean string x.

Most researchers believe that NP does *not* have polynomial-sized circuits; i.e., NP is *hard* for polynomial-sized circuits. One might then ask *how* hard NP is for polynomial circuits. One way of viewing this to question is to ask on how large a fraction of the inputs in $\{0,1\}^n$ can a polynomial-sized circuit compute a given NP function. We say that f is "$(1-\delta)$-hard for polynomial circuits" is for every family $\langle C_n \rangle$ of polynomial-sized circuits, $\mathbf{P}[f(x) = C_n(x)] \leq 1 - \delta$. Note that asserting NP is hard for polynomial circuits is the same as saying that there is a function $f \in \mathsf{NP}$ which is $(1 - 2^{-n})$-hard for polynomial circuits. Also note that no function is $(1-\delta)$-hard for $\delta \geq 1/2$ because either the circuit which always outputs 1 or the circuit that always outputs 0 gets f right on at least half of all inputs. Under the assumption that NP does not have polynomial circuits, it is of interest to know just how hard NP is in this sense.

In [O02], the second author addresses this question. Starting from the assumption that there is a function in NP which is $(1 - 1/n^{O(1)})$-hard for polynomial circuits, [O02] shows the existence of a function in NP which is $(1/2 + n^{-1/2+\delta})$-hard for polynomial circuits (for any small $\delta > 0$). The main technical theorem in [O02] is that if f is a balanced function which is $(1-\delta)$-hard for polynomial circuits, and g is a function satisfying $Z(g, 1-2\delta) \leq \eta$, then $g \otimes f$ is essentially $(\frac{1}{2} + \frac{1}{2}\sqrt{\eta})$-hard for polynomial circuits.

In order to apply this technical theorem to convert a slightly hard function in NP to a very hard function in NP, it is necessary to ensure that $g \otimes f \in \mathsf{NP}$ when $f \in \mathsf{NP}$. Recall that NP is the class of functions f which have easily verified proofs of $f = 1$. In order for $g \otimes f$ to have easily verified proofs of $g \otimes f = 1$, it suffices for g to be (a) in NP, and (b) monotone. For in this case, we can prove that $g \otimes f = 1$ by proving that some subset of the inputs to g are 1, and each of these is a statement of the form $f = 1$, which has an easily verified proof because $f \in \mathsf{NP}$.

Hence to amplify hardness within NP, [O02] needs to find a monotone function in NP such that $Z(g, 1 - 1/n^{O(1)})$ is very small. This exact problem is addressed in the present paper. Take g to be the function from Theorem 7.1 on k inputs. This function is easily seen to be in P, hence in NP. If we pick $k = n^C$ and $\epsilon = n^c/k$ for some constants C and c, then Theorem 7.1 tells us that $Z(g, 1 - 1/\tilde{\Omega}(n^{c/2})) \leq 1/k^{1-c/C}$. Hence if f $(1 - 1/n^{O(1)})$-hard for polynomial circuits, by choosing c and C sufficiently large, we can arrange for $g \otimes f$ — which has input length $kn = n^{C+1}$ — to be $(1/2 + (kn)^{-1/2+\delta})$-hard for any small $\delta > 0$. This is the result of [O02].

Note that Theorem 1.4 is not useful in this context, since the amplifying function g must be in NP, and Talagrand's function is not even explicit.

2.3 Neural networks

In the theory of neural networks (see e.g. [H99] for background), a neuron is modeled as a weighted majority function. For physical and biological reasons, it is expected that such a function would be noise stable. In [BKS98] it is shown that there exists a universal constant C such that for all weighted majority functions M, $Z(M, 1-\epsilon) \geq 1 - C\epsilon^{1/4}$. Peres [Pe98] has improved this to $1 - C\epsilon^{1/2}$.

If we consider the simplest kind of neural network, in which every variable and every majority output is read only once, we obtain a tree circuit of weighted majority gates. Using a simple exchange of variables, we may assume that all the weights of the majority functions are positive and hence that the network represents a monotone function. Proposition 1.1 implies that the network is insensitive to noise rate of $n^{-\alpha}$ for $\alpha > 1/2$, where n is the number of inputs to the function. Our construction in Theorem 1.2 implies on the other hand that this is tight, i.e., for every $\alpha < 1/2$, there exists a neural network in which every variable and every output is read once, and the network is sensitive to noise rate $n^{-\alpha}$.

2.4 Sensitivity of election schemes

One of the desired properties of election schemes is robustness. Consider the following simple model: There are n voters who have to decide between candidate -1 and candidate 1. Suppose that voter i wants to vote x_i, and that the x_i's are uniformly random and independent. Suppose furthermore that due to confusion and some technical errors, the vote of voter i is recorded as y_i where $\mathbf{P}[x_i = y_i] = 1 - \epsilon$ independently for all i. In this setting it is natural to require that the vote outcome $f(y_1, \ldots, y_n)$ be governed by a symmetric balanced monotone function. Moreover, if we want to minimize the effect of the confusion and errors, we want to maximize $\mathbf{P}[f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)] = Z(f, 1 - 2\epsilon)$.

Let us compare two election schemes. In the first scheme, f is the simple majority function. Here $Z(f, 1-2\epsilon)$ is of order $1 - \epsilon^{1/2}$. In the second scheme, we have a two level majority function; e.g., each state votes by simple majority for an elector, and the majority of the electors' votes chooses the president. Here, if we assume $n^{1/2}$ electors, a calculation as in the proof of Theorem 1.2 shows that $Z(f, 1-2\epsilon)$ is of order $\epsilon^{1/4}$. Hence the "electoral college" system is much more sensitive to noise. In fact, Theorem 1.2 suggests that adding more levels of subelectors (such as voting by county first) increases the sensitivity of the election to noise, up to its maximum possible level for a monotone function.

3 Sensitivity of majorities

3.1 Majority

We denote the majority function on k bits by MAJ_k. Using asymptotic results for random walks, one can prove (cf. [O02]):

Proposition 3.1 *For every $\eta \in [-1, 1]$,*

$$|Z(\mathsf{MAJ}_k, \eta) - \frac{2}{\pi} \arcsin(\eta)| \le O(1/\sqrt{k}).$$

Much more can be said when η is very close to 1, specifically, when $1 - \eta$ is small compared to $1/k$. For η close to 1, we prefer to view $Z(f, \eta)$ in terms of the probability that flipping input bits of f flips the output bit. We use the following lemma in the proof of Theorem 1.3.

Lemma 3.2 *Suppose $k \ge 3$ and $\delta \le 1/k$. Say we pick a random input to MAJ_k — call it x — and then construct y by flipping each bit of x independently with probability δ. Then*

$$\mathbf{P}[\mathsf{MAJ}_k(x) \ne \mathsf{MAJ}_k(y)] \ge \sqrt{\frac{2}{\pi}} \sqrt{k}\ \delta\ \exp(-1/3k) \exp(-\delta k).$$

Proof: Clearly,

$$\begin{aligned} &\mathbf{P}[\mathsf{MAJ}_k(x) \ne \mathsf{MAJ}_k(y)] \\ &\qquad \ge \mathbf{P}[\mathsf{MAJ}_k(x) \ne \mathsf{MAJ}_k(y) | \text{exactly one flip}] \times \mathbf{P}[\text{exactly one flip}], \end{aligned} \tag{6}$$

and $\mathbf{P}[\text{exactly one flip}] = k\delta(1-\delta)^{k-1}$. By elementary calculus, $(1-\delta)^{k-1} \ge \exp(-\delta k)$ for $\delta \le 1/k$. Therefore,

$$\mathbf{P}[\text{exactly one flip}] = k\delta(1-\delta)^{k-1} \ge k\delta \exp(-\delta k). \tag{7}$$

The probability that the majority flips given that there is exactly one flipped bit in x, is exactly the probability that the remaining input bits split evenly — i.e.,

$$\begin{aligned} \mathbf{P}[\mathsf{MAJ}_k(x) \ne \mathsf{MAJ}_k(y) | \text{exactly one flip}] &= \binom{k-1}{(k-1)/2} 2^{-(k-1)} \\ &\ge \sqrt{\frac{2}{\pi k}}(1 - 1/4k) \ge \sqrt{\frac{2}{\pi k}} \exp(-1/3k), \end{aligned} \tag{8}$$

where the first inequality follows by Stirling's formula and the second since $1 - 1/4k \le \exp(-1/3k)$ for $k \ge 3$. Combining (6), (7) and (8) we obtain the required result. □

3.2 Recursive majority

We begin with a formal definition of the recursive majority function.

Definition 3.3 *For $f : \Omega_n \to \{-1, +1\}, g : \Omega_m \to \{-1, +1\}$, we let $f \otimes g$ denote the function $f \otimes g : \Omega_{nm} \to \{-1, +1\}$ defined by*

$$f \otimes g\ (x_1, \ldots, x_{nm}) = f\left(g(x_1, \ldots, x_m), \ldots, g(x_{(n-1)m+1}, \ldots, x_{nm})\right)$$

For ℓ an integer, we define $f^{\otimes^\ell} = f$ if $\ell = 1$, and $f^{\otimes^\ell} = f \otimes (f^{\otimes^{\ell-1}})$ otherwise. We let $\mathsf{REC\text{-}MAJ\text{-}}t_\ell = \mathsf{MAJ}_t^{\otimes^\ell}$.

The following proposition is immediate, yet useful.

Proposition 3.4 *If g is a balanced function and f is any function, then $Z(f \otimes g, \eta) = Z(f, Z(g, \eta))$.*

In this section we prove Theorem 1.2. It is easy to calculate (and well known) that for the majority function on $k = 2r+1$, MAJ_k,

$$II(\mathsf{MAJ}_k) = \frac{2r+1}{2^{4r}} \binom{2r}{r}^2, \qquad I(\mathsf{MAJ}_k) = \frac{2r+1}{2^{2r}} \binom{2r}{r}.$$

Note therefore that $I(\mathsf{MAJ}_k) \to \sqrt{2/\pi}\sqrt{k}$ as $k \to \infty$. Hence Theorem 1.2 follows almost immediately from the following proposition:

Proposition 3.5 *Let $f : \Omega_k \to \{-1, +1\}$ be a balanced function, and let*

$$a = \sum_S |S| \hat{f}^2(S), \qquad b = \sum_{|S|=1} \hat{f}^2(S).$$

(Note that $a = I(f)$, and if f is monotone, $b = II(f)$.) If $a > 1$ and $b < 1$, then $Z(f^{\otimes^\ell}, 1-\delta) \le \epsilon$, for $\ell \ge \Big(\log_a(1/\delta) + \log_{1/b}(1/\epsilon)\Big)(1 + r(\epsilon, \delta))$, where $r(\epsilon, \delta) \to 0$ as $\epsilon \to 0$ and $\delta \to 0$.

Proof: (sketch) Let $f = \sum_S \hat{f}^2(S) u_S$ be the Fourier expansion of f. Letting $p(\eta) := Z(f, \eta) = \sum_S \hat{f}^2(S) \eta^{|S|}$, we see that $p(\eta)$ is a convex polynomial function of η which satisfies

$$p(0) = 0, \quad p(1) = 1, \quad p'(0) = \textstyle\sum_{|S|=1} \hat{f}^2(S) = b, \quad p'(1) = \sum_S |S| \hat{f}^2(S) = a. \tag{9}$$

Proposition 3.4 implies that

$$Z(f^{\otimes^\ell}, \eta) = p^{(\ell)}(\eta) := \underbrace{p(p(\cdots p(\eta) \cdots))}_{\ell \text{ times}}. \tag{10}$$

The claim of the proposition now follows by standard arguments on iterations of convex functions (for more details, see the long version of this paper). □

4 Sensitivity to small noise

In this section we prove Theorem 1.3. We do this by proving

Theorem 4.1 *There exists an explicit infinite family of balanced monotone functions $f_n : \Omega_n \to \{-1, +1\}$ with the following property:*

$$Z(f_n, 1 - \epsilon/M) \le 1 - \epsilon + O(\epsilon^2),$$

where $M = \sqrt{n}/\Theta(\log^t n)$, and $t = \log_2 \sqrt{\pi/2} = .3257\ldots$

Proof of Theorem 1.3: Let f_n be the function constructed at Theorem 4.1, and let ϵ be such that $Z(f_n, 1-\epsilon/M) \leq 1-\epsilon+O(\epsilon^2) < 1-\delta'+o(1)'$, where $\delta' > 0$. Let $g = \mathsf{REC\text{-}MAJ\text{-}3}_\ell$ where ℓ is chosen is such a way that $Z(g, 1-\delta'/2) \leq 1-\delta$ (such ℓ exists by Theorem 1.2). Taking $g_n = g \otimes f_n$, we obtain the desired result. □

The construction in Theorem 4.1 again consists of recursive majorities, where now the number of inputs to the majority varies with the level. The estimates on the sensitivity of these majority functions are derived via Lemma 3.2.

Proof of Theorem 4.1: Since we are dealing with correlations close to 1, it will be more helpful to look at their difference from 1. In particular, we will prove the following equivalent formulation of the theorem: Let x be a randomly chosen input to f_n, and suppose we flip each bit of x independently with probability ϵ/M, forming y. Then the probability that $f_n(x) = f_n(y)$ is at least $\epsilon - O(\epsilon^2)$.

The function $f = f_n$ will be given by recursive majorities of increasing arity: $f_n = \mathsf{MAJ}_{k_1} \otimes \mathsf{MAJ}_{k_2} \otimes \cdots \otimes \mathsf{MAJ}_{k_\ell}$. We will select $k_i = 3^{2^{i-1}+1}$, so "from the top down" the majorities have arity 9, 27, 243, etc. Note that $k_{i+1} = k_i^2/3$. With these choices, the number of inputs is $n = 3^{2^\ell+\ell-1}$. Hence $\ell \leq \log_2 \log_3 n$.

Let $\delta_0 = \epsilon/M$, and recursively define δ_{i+1} to be the probability that the output of a $\mathsf{MAJ}_{k_{\ell-i}}$ flips, given that each of its inputs is flipped independently with probability δ_i. Since all MAJ functions are balanced, Proposition 3.4tells us that the probability that the output of f is flipped is δ_ℓ. We will show that $\delta_\ell \geq \epsilon - O(\epsilon^2)$.

By Lemma 3.2,

$$\delta_{i+1} \geq g(k_{\ell-i}) \exp(-\delta_i k_{\ell-i}) \delta_i,$$

where:

$$g(t) := \frac{1}{\sqrt{\pi/2}} \sqrt{k_{\ell-i}} \exp(-1/3k_{\ell-i}).$$

Recursively define $\eta_0 = \eta'_0 = \delta_0$, and:

$$\eta_{i+1} = g(k_{\ell-i}) \exp(-\eta_i k_{\ell-i}) \eta_i, \quad \eta'_{i+1} = g(k_{\ell-i}) \eta'_i.$$

Since the probability that the output of MAJ flips is an increasing function of δ, we can conclude that $\delta_i \geq \eta_i$ for every i. But clearly $\eta'_i \geq \eta_i$ for every i. Hence, for every i, $\eta_{i+1} \geq g(k_{\ell-i}) \exp(-\eta'_i k_{\ell-i}) \eta_i$. It follows immediately that:

$$\begin{aligned}
\eta_\ell &\geq \Big(\prod_{i=0}^{\ell-1} g(k_{\ell-i}) \exp(-\eta'_i k_{\ell-i})\Big) \eta_0 \\
&= \Big(\frac{1}{\sqrt{\pi/2}}\Big)^\ell \prod_{j=1}^{\ell} \sqrt{k_j}\ \exp(-\frac{1}{3}\sum_{j=1}^{\ell} k_j^{-1}) \cdot \exp\Big[\sum_{i=0}^{\ell-1} -\eta'_i k_{\ell-i}\Big] \cdot \delta_0
\end{aligned}$$

Defining

$$\begin{aligned}
M := \prod_{m=1}^{\ell} g(k_m) = \Big(\frac{1}{\sqrt{\pi/2}}\Big)^{\log_2 \log_3 n} &\exp(-\frac{1}{3}\sum_{j=1}^{\ell} k_j^{-1}) \\
&= \Big(\frac{1}{\sqrt{\pi/2}}\Big)^{\log_2 \log_3 n} \sqrt{n} \exp(-O(1)),
\end{aligned}$$

and $\delta_0 := \epsilon/M$, we obtain

$$\eta_\ell \geq M \cdot \exp\Big[\sum_{i=0}^{\ell-1} -\eta'_i k_{\ell-i}\Big] \cdot (\epsilon/M) = \epsilon \cdot \exp\Big[\sum_{i=0}^{\ell-1} -\eta'_i k_{\ell-i}\Big].$$

Since $\delta_\ell \geq \eta_\ell$, it remains to show:

$$\exp\Big[\sum_{i=0}^{\ell-1} -\eta'_i k_{\ell-i}\Big] \geq 1 - O(\epsilon).$$

By the recursive definition of η'_i, we immediately have $\eta'_i = (\prod_{j=0}^{i-1} g(k_{\ell-j}))\eta'_0$. Hence $\eta'_i = M(\prod_{m=1}^{\ell-i} g(k_m)^{-1})\eta'_0 = \epsilon(\prod_{m=1}^{\ell-i} g(k_m)^{-1})$. Therefore:

$$\exp\Big[\sum_{i=0}^{\ell-1} -\eta'_i k_{\ell-i}\Big] = \exp\Big[-\epsilon \sum_{m=1}^{\ell} \frac{k_m}{g(k_1)g(k_2)\cdots g(k_m)}\Big].$$

Hence if we can show $\sum_{m=1}^{\ell} k_m/g(k_1)g(k_2)\cdots g(k_m) = O(1)$ then we're done. The first term in this sum is $k_1/g(k_1) = O(1)$. The ratio of the mth term to the $(m-1)$th term is $k_m/k_{m-1}g(k_m)$. But $k_{m-1} = \sqrt{3}\sqrt{k_m}$ by definition, so this ratio is $\sqrt{k_m}/\sqrt{3}g(k_m) = \sqrt{\pi/2}/\sqrt{3}\exp(-1/3k_m) < 1$. Hence the terms in the sum decrease geometrically, so the sum is indeed $O(1)$. $\square$

5 Talagrand's function

In [T96], Talagrand gives a randomized construction of a monotone $f_n : \Omega_n \to \{-1, +1\}$ with the following property: at least an $\Omega(1)$ fraction of points x in Ω_n satisfy both $f_n(x) = -1$, and $\#\{x' : \Delta(x, x') = 1 \text{ and } f(x) = +1\} \geq \Omega(n^{1/2})$, where Δ denotes Hamming distance. It is natural to conjecture that this function is sensitive to slight $n^{-1/2}$ noise, as we prove below.

Talagrand's function $f = f_n$ is a random CNF on its n inputs. Specifically, f is the $2^{\sqrt{n}}$-wise AND of $\sqrt{n}$-wise ORs, where each OR's inputs are selected independently and uniformly at random (with replacement) from $[n]$. To prove Theorem 1.4, it suffices to prove that if we pick f, x, and $x' := N_\epsilon(x)$ at random (where $\epsilon = n^{-1/2}$), then:

$$\mathbf{E}_f\big[\mathbf{P}[f(x) \neq f(N_\epsilon(x))]\big] \geq \Omega(1).$$

Proof of Theorem 1.4: (sketch)

$$\begin{aligned}\mathbf{E}_f\big[\mathbf{P}[f(x) \neq f(N_\epsilon(x))]\big] &= \mathbf{E}_{x,\,x'}\big[\mathbf{P}_f[f(x) \neq f(x')]\big] \\ &= 2\mathbf{E}_{x,\,x'}\big[\mathbf{P}_f[f(x) = -1, f(x') = +1]\big], \quad (11)\end{aligned}$$

by symmetry, since x and x' have the same distribution. We want to show that $(11) \geq \Omega(1)$.

Fix x and x'. Let $n_{+\star}$ denote the number of indices on which x is $+1$, let $n_{\star+}$ denote the number of indices on which x' is $+1$, and let n_{++} denote the number of indices on which *both* x and x' are $+1$.

Since f has a fairly simple form — the AND of ORs, where the ORs' inputs are completely independent — it is easy to write $\mathbf{P}_f[f(x) = -1, f(x') = +1]$ explicitly in terms of $n_{+\star}$, $n_{\star+}$, and n_{++}:

$$\mathbf{P}_f[f(x) = -1, f(x') = +1] = p_{\star-}^{2^{\sqrt{n}}} - p_{--}^{2^{\sqrt{n}}}, \tag{12}$$

where

$$\begin{aligned} p_{\star-} &= 1 - \big(\frac{n_{\star+}}{n}\big)^{\sqrt{n}}, \\ p_{--} &= 1 - \big(\frac{n_{\star+}}{n}\big)^{\sqrt{n}} - \big(\frac{n_{+\star}}{n}\big)^{\sqrt{n}} + \big(\frac{n_{++}}{n}\big)^{\sqrt{n}}. \end{aligned}$$

By the mean value theorem, (12) is bounded from below by:

$$2^{\sqrt{n}}(p_{\star-} - p_{--})p_{--}^{2^{\sqrt{n}}}. \tag{13}$$

Now $n_{+\star} \sim \text{Binomial}(n, 1/2)$, and similarly for $n_{\star+}$. Hence for sufficiently large n, both quantities are in the range $[n/2 - \sqrt{n}, n/2 + \sqrt{n}]$, except with probability .05. Also, $n_{++} \sim \text{Binomial}(n_{+\star}, 1-\epsilon)$, so for sufficiently large n and if $\epsilon \geq n^{-1/2}$, n_{++} is no larger than $(1 - \epsilon + 2\sqrt{\epsilon/n_{+\star}})n_{+\star}$, except with probability .05. Taking all these facts together via a union bound, we may conclude that except with probability .15,

$$n_{+\star} \in \left[\frac{n}{2} - \sqrt{n}, \frac{n}{2} + \sqrt{n}\right], \quad n_{\star+} \in \left[\frac{n}{2} - \sqrt{n}, \frac{n}{2} + \sqrt{n}\right], \quad \frac{n_{++}}{n_{\star+}} \leq 1 - \epsilon + 3\sqrt{\frac{\epsilon}{n}}. \tag{14}$$

We would like to show that $\mathbf{E}_{x,x'}[(13)] \geq \Omega(1)$. Since (14) happen with probability at least .85, it suffices to prove $\mathbf{E}_{x,x'}[(13)] \geq \Omega(1)$ *conditioned on* these three events holding. But in this case,

$$\begin{aligned} \mathbf{E}_{x,x'}[(13)] &= 2^{\sqrt{n}}(p_{\star-} - p_{--})\Big[1 - \big(\frac{n_{\star+}}{n}\big)^{\sqrt{n}} - \big(\frac{n_{+\star}}{n}\big)^{\sqrt{n}} + \big(\frac{n_{++}}{n}\big)^{\sqrt{n}}\Big]^{2^{\sqrt{n}}} \\ &\geq 2^{\sqrt{n}}(p_{\star-} - p_{--})\Big[1 - \big(\frac{n_{\star+}}{n}\big)^{\sqrt{n}} - \big(\frac{n_{+\star}}{n}\big)^{\sqrt{n}}\Big]^{2^{\sqrt{n}}} \\ &\geq 2^{\sqrt{n}}(p_{\star-} - p_{--})\Big[1 - (1/2 + n^{-1/2})^{\sqrt{n}} - (1/2 + n^{-1/2})^{\sqrt{n}}\Big]^{2^{\sqrt{n}}} \\ &\geq 2^{\sqrt{n}}(p_{\star-} - p_{--})[1 - 2e/2^{\sqrt{n}}]^{2^{\sqrt{n}}} \\ &\geq e^{-2e}2^{\sqrt{n}}(p_{\star-} - p_{--}) \\ &= e^{-2e}\Big(2\frac{n_{\star+}}{n}\Big)^{\sqrt{n}}\Big(1 - \Big(\frac{n_{++}}{n_{\star+}}\Big)^{\sqrt{n}}\Big) \\ &\geq e^{-2e}(1 - 2n^{-1/2})^{\sqrt{n}}\Big(1 - \Big(\frac{n_{++}}{n_{\star+}}\Big)^{\sqrt{n}}\Big) \\ &\geq e^{-2e-2}\Big(1 - \Big(\frac{n_{++}}{n_{\star+}}\Big)^{\sqrt{n}}\Big) \\ &\geq e^{-2e-2}\Big(1 - (1 - \epsilon + 2\sqrt{\epsilon/n})^{\sqrt{n}}\Big). \end{aligned} \tag{15}$$

When $\epsilon = n^{-1/2}$, the quantity $(1-\epsilon+2\sqrt{\epsilon/n})^{\sqrt{n}}$ exceeds e^{-1}. Hence (15) is at least $e^{-2e-2} \geq \Omega(1)$, and we're done. $\square$

6 Tribes and high sensitivity

We have mostly settled the question of how small ϵ can be, such that there is a monotone function f satisfying $Z(f, 1-\epsilon) \leq 1-\Omega(1)$. At the other end of the spectrum, one might ask: given an initial correlation $\delta < 1-\Omega(1)$, which monotone function f makes $Z(f,\delta)$ as close to 0 as possible? A nearly optimal function for this problem (which is tight to within a constant factor if the initial correlation δ is small enough) is the so-called tribes function of Ben-Or and Linial [BL90].

Let AND_k denote the And function on k bits (i.e., $\mathsf{AND}_k(x) = -1$ iff $x_i = -1$ for all $1 \leq i \leq k$), and let OR_k denote the Or function on k bits. For each $b \in \mathbf{N}$, define $n = n_b$ to be the smallest integral multiple of b such that $(1-2^{-b})^{n/b} \leq 1/2$, so n is very roughly $(\ln 2)b2^b$, and $b = \lg n - \lg\ln n + o(1)$. (Here $\lg n$ denotes $\log_2 n$.) Now define the tribes function T_n to be $\mathsf{OR}_{n/b} \otimes \mathsf{AND}_b$. This function is monotone, and by construction it's near-balanced; it's easy to see that $\mathbf{P}[T_n = +1] = (1-2^{-b})^{n/b} = 1/2 - O(\log n/n)$.

One can calculate $Z(T_n, \eta)$ directly and exactly:

Proposition 6.1 $\quad Z(T_n,\eta) = 1-4\big[(1-2^{-b})^{n/b} - (1-(2-(\frac{1}{2}+\frac{1}{2}\eta)^b)2^{-b})^{n/b}\big].$

Corollary 6.2 $\quad Z(T_n,\eta) \leq (1+o(1))\frac{\lg^2 n}{n}\eta(1+\eta)^b + O(\log^2 n/n^2).$
Therefore if $\eta \leq O(1/\log n)$, then $Z(T_n,\eta) \leq O(\eta \log^2 n/n)$.

We omit the proofs of these results from this extended abstract. A similar result to Corollary 6.2 appears in [O02], with a more complicated proof.

Now we give a monotone function for which $Z(f,\delta)$ is small when $\delta \leq 1-\Omega(1)$.

Theorem 6.3 *Let $\delta \leq 1-\Omega(1)$. Then there is an infinite family of monotone functions $\{g_n\}$ satisfying:*

$$Z(g_n,\delta) \leq \frac{\log^{1+u'} n}{n},$$

where u' is any number exceeding $u = \log_{4/3} 3 = 3.818\ldots$.

Proof: The idea is to first use REC-MAJ-3 to reduce δ to $\eta := 1/\log n$; then, apply a tribes function.

Let T_n be any tribes function. We will construct $g_{n'}$ on $n' := n\log^{u'} n$ inputs. Let ℓ be the REC-MAJ-3 depth necessary from Theorem 1.2 to reduce δ correlation down to $1/\log n$ correlation. Hence $\ell = (1+o(1))\log_{4/3}(\log n)$ (since $1-\delta \geq \Omega(1)$). Put $h = \text{REC-MAJ-}3_\ell$, so h is a function on $3^\ell = \log^{u'} n$ inputs. Let $g_{n'} = T_n \otimes h$.

By construction, $Z(h,\delta) \leq 1/\log n$. By Corollary 6.2, $Z(T_n, 1/\log n) \leq O(\log n/n)$. Since h is balanced, by Proposition 3.4 we get $Z(g_{n'},\delta) \leq O(\log n/n)$. The result follows, since as a function of n', $O(\log n/n)$ is $\log^{1+u'} n'/n'$ (taking u' slightly larger to kill any constant factors). $\square$

As we can see from the following proposition, when the initial correlation $0 < \delta < 1$ is a constant, the above result is tight up to a factor of $\log^{2.818} n$:

Proposition 6.4 *If $f : \Omega_n \to \{-1,+1\}$ is monotone, then $Z(f,\eta) \geq \Omega(\eta \log^2 n/n)$.*

Proof:

$$Z(f,\eta) = \sum_S \eta^{|S|} \hat{f}^2(S) \geq \eta \sum_{|S|=1} \hat{f}^2(S) \geq \Omega(\eta \log^2 n/n),$$

by a result of [KKL88] (using the fact that f is monotone). □

It also follows from this proposition and Corollary 6.2 that when the initial correlation η is $O(1/\log n)$, the tribes function by itself is maximally sensitive among monotone functions, to within a constant factor.

7 High sensitivity to small noise, and Fourier concentration around $\sqrt{n}$

It seems natural to combine the functions from Theorems 1.3 and 6.3, via Proposition 3.4. One gets:

Theorem 7.1 *There exists an explicit infinite family of monotone functions $f_n : \Omega_n \to \{-1,+1\}$ with the following property: $Z(f_n, 1-1/Q) \leq \epsilon$, where:*

$$Q = \frac{\sqrt{n\epsilon}}{(\log(n\epsilon))^t \log(1/\epsilon)^{(1+u')/2}},$$

$t = .3257\ldots$, and $(1+u')/2 = 2.409\ldots$.

Using the relationship $Z(f,\eta) = \sum_S \eta^{|S|} \hat{f}^2(S)$, it's easy to conclude:

Corollary 7.2 *There exists an explicit infinite family of monotone functions $f_n : \Omega_n \to \{-1,+1\}$ satisfying:*

$$\sum_{|S| \leq Q} \hat{f_n}^2(S) \leq \epsilon,$$

where $Q = \tilde{\Omega}(\sqrt{n\epsilon})$ is the quantity from Theorem 7.1.

From (3), one can easily derive the well-known fact that for all monotone $f : \Omega_n \to \{-1,+1\}$, $\sum_{|S| \leq \epsilon^{-1}\sqrt{n}} \hat{f}^2(S) \geq 1 - \epsilon$. That is, every monotone function has almost all the ℓ_2 mass of its Fourier spectrum concentrated on coefficients of degree up to $O(\sqrt{n})$. Corollary 7.2 demonstrates that this bound is tight up to polylog factors.

Acknowledgment: We would like to thank Gil Kalai for providing encouragement to write this result and Yuval Peres for interesting discussions.

References

[AN72] K. Athreya, P. Ney. Branching processes. Springer-Verlag, New York-Heidelberg, 1972.

[BDG88] J. Balcázar, J. Díaz, J. Gabarró. Structural Complexity I, II. Springer-Verlag, Heidelberg, 1988.

[BJT99] N. Bshouty, J. Jackson, T. Tamon. Uniform-distribution attribute noise learnability. *Workshop on Computational Learning Theory*, 1999.

[BKS98] I. Benjamini, G. Kalai, O. Schramm. Noise sensitivity of boolean functions and applications to percolation. Preprint.

[BL90] M. Ben-Or, N. Linial. Collective coin flipping. In *Randomness and Computation*, S. Micali ed. Academic Press, New York, 1990.

[BT96] N. Bshouty, C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM* 43(4), 1996.

[DK00] D.-Z. Du, K.-I Ko. Theory of Computational Complexity. Wiley Interscience, New York, 2000.

[F98] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica* 18(1), 1998, 27–36.

[FK96] E. Friedgut, G. Kalai. Every monotone graph property has a sharp threshold. *Proc. Amer. Math. Soc.* 124, 1996, 2993–3002.

[H99] S. Haykin. Neural Networks, 2nd Edition. Prentice Hall, 1999.

[J97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3), 1997.

[KKL88] J. Kahn, G. Kalai, N. Linial. The influence of variables on boolean functions. *Foundations of Computer Science*, 1988.

[KOS02] A. Klivans, R. O'Donnell, R. Servedio. Learning intersections and thresholds of halfspaces. To appear.

[LMN93] N. Linial, Y. Mansour, N. Nisan. Constant depth circuits, Fourier transform, and learnability. *J. Assoc. Comput. Mach.* 40, 1993, 607–620.

[M94] Y. Mansour. Learning boolean functions via the Fourier transform. *Theoretical Advances in Neural Computing and Learning*, Kluwer Acad. Publ., Dordrecht (1994), 391–424.

[M98] E. Mossel. Recursive reconstruction on periodic trees. *Random Structures Algorithms*, 13, 1998, no. 1, 81–97.

[O02] R. O'Donnell. Hardness amplification within NP. Symposium on the Theory Of Computation, 2002.

[Pa93] C. Papadimitriou. Computational Complexity. Addison Wesley, Reading, MA, 1993.

[Pe98] Y. Peres. Personal communication, 1998.

[T96] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica* 16, 1996, no. 2, 243–258.

[V84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 40, 1994, no. 2, 445–474.

Elchanan Mossel
Hebrew University of Jerusalem and Microsoft Research
mossel@microsoft.com
Ryan O'Donnell
MIT Mathematics Department
odonnell@theory.lcs.mit.edu