Hardness Amplification Within NP

Ryan O'Donnell*
MIT Mathematics Department
Cambridge, MA 02139-3594
email: odonnell@theory.lcs.mit.edu

June 20, 2002

Abstract

In this paper we investigate the following question: If NP is slightly hard on average, is it very hard on average? We show the answer is yes; if there is a function in NP which is infinitely often balanced and $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then there is a function in NP which is infinitely often $(\frac{1}{2} + n^{-1/2+\epsilon})$-hard for circuits of polynomial size. Our proof technique is to generalize the Yao XOR Lemma, allowing us to characterize nearly tightly the hardness of a composite function $g(f(x_1), \ldots, f(x_n))$ in terms of: (i) the original hardness of f, and (ii) the *expected bias* of the function g when subjected to random restrictions. The computational result we prove essentially matches an information-theoretic bound.

1 Introduction

Supposing that NP is hard for polynomial circuits, is it hard on average? That is, does it contain functions for which every polynomial-sized circuit outputs the wrong answer on close to half of all inputs? Let us make the following definition:

Definition 1 *A boolean function $f : \{0,1\}^n \to \{0,1\}$ is* $(1 - \delta)$-hard for circuits of size s *if no circuit of size s can correctly compute f on a $1 - \delta$ fraction of the inputs $\{0,1\}^n$.*

Of course, we don't know whether there is a function in NP which is even $(1 - 2^{-n})$-hard for polynomial circuits, since this is precisely the NP vs. P/poly problem. However, under the reasonable assumption that NP is at least slightly hard in the above sense, it is of interest to know just *how* hard it really is. This problem has been extensively studied for EXP in the context of derandomization, and very strong results are known — see [BFNW93, Imp95, IW97, STV01].

A crucial ingredient for some of these results is the Yao XOR Lemma. This is a kind of "direct product theorem", which describes what happens to the hardness of a function f when one takes many independent copies of it and applies another function, g, to the results. In Yao's case [Yao82], $g = \mathsf{PARITY}$ and roughly speaking one gets that the resulting function $f \oplus f \oplus \cdots \oplus f$ is exponentially more hard than f.

While this is very useful for amplifying hardness within EXP, it doesn't shed any light on the hardness of NP. The reason is that $f \oplus f \oplus \cdots \oplus f$ is not necessarily in NP, even when f is. In order to ensure the composite function is in NP, we must use a g which is *monotone* (and also in NP).

*Research partially supported by NSERC fellowship PGSA-221401-99.

We are thus led to the question of whether there is a monotone function which amplifies hardness.

In this paper we investigate the direct product question in a general form:

Definition 2 *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^k \to \{0,1\}$ be boolean functions. Then we define $g \otimes f$ to be the boolean function $(\{0,1\}^n)^k \to \{0,1\}$ given by $(x_1, \ldots, x_k) \mapsto g(f(x_1), \ldots, f(x_k))$.*

Question 1 *Suppose $f : \{0,1\}^n \to \{0,1\}$ is a balanced boolean function which is $(1-\delta)$-hard for circuits of size s. Let $g : \{0,1\}^k \to \{0,1\}$ be any function. What is the hardness of the composite function $g \otimes f$?*

By a *balanced* function we simply mean one which is 1 on exactly half of all inputs. We consider only balanced functions f for technical reasons that will become clear later.

The tight answer we give to the above question is in terms of the *expected bias of g*, which we now define. Let $h : \{0,1\}^n \to \{0,1\}$ be a boolean function.

Definition 3 *The* bias *of h is* $\mathrm{bias}(h) = \max\{\Pr[h=0], \Pr[h=1]\}$.

(Here and throughout, we view functions' domains as probability spaces with the uniform measure.)

Definition 4 *We denote by P_δ^n the set of random restrictions ρ on n coordinates, in which each coordinate is mapped independently to $\star$ with probability δ, 0 with probability $(1-\delta)/2$, and 1 with probability $(1-\delta)/2$. We write h_ρ for the function given by applying restriction ρ to function h.*

Our central definition:

Definition 5 *The* expected bias of h at δ *is:*

$$\mathrm{ExpBias}_\delta(h) = \mathop{\mathbf{E}}_{\rho \in P_\delta^n}[\mathrm{bias}(h_\rho)].$$

With this definition in place, we can now answer Question 1:

Answer 1 *Roughly speaking, the hardness of $g \otimes f$ is* $\mathrm{ExpBias}_{2\delta}(g)$. *To state our hardness theorem exactly:*

Theorem 1 *For every $r > 0$, the function $g \otimes f$ is $(\mathrm{ExpBias}_{(2-r)\delta}(g) + \epsilon)$-hard for circuits of size $s' = \Omega(\frac{\epsilon^2/\log(1/\delta)}{k}\, s)$.*

In Section 2 we explain from an intuitive, information-theoretic perspective why Answer 1 should be the correct answer; then in Section 3 we transfer these ideas to the circuit setting and give the computational proof of Theorem 1.

Note that a form of Yao's XOR Lemma follows as a corollary; since $\mathrm{ExpBias}_\delta(\mathsf{PARITY}_k)$ is easily calculated to be $\frac{1}{2} + \frac{1}{2}(1-\delta)^k$, we get (taking $r = 1$):

Corollary 1 *(Yao) If f is a balanced boolean function which is $(1-\delta)$-hard for circuits of size s, then $f \oplus f \oplus \cdots \oplus f$ (k times) is $(\frac{1}{2} + \frac{1}{2}(1-\delta)^k + \epsilon)$-hard for circuits of size $\Omega(\frac{\epsilon^2/\log(1/\delta)}{k}\, s)$.*

If we want to do hardness amplification within NP, we are led to look for monotone functions g such that $\mathrm{ExpBias}_{2\delta}(g)$ is much smaller than $1-\delta$. Qualitatively, we want a monotone function which: (i) is nearly balanced, and (ii) remains close to balanced even when subjected to random restrictions. For technical reasons, it turns out that functions for which each input variable has small "influence" do well. Thus we are led to consider a classic pair of monotone functions introduced by Ben-Or and Linial [BL90] which have this property.

The first of these is the "recursive majority of 3" function. This function does especially well when δ is very small, and can take $(1-1/\mathrm{poly}(n))$-hardness to $(1/2+n^{-\alpha})$-hardness, for a certain small $\alpha > 0$. The second function is the "tribes" function, and it gets very close to $1/2$ when δ starts out fairly large. By first getting hardness down to $1/2+o(1)$ via recursive majority, and then applying the tribes function, we get the main theorem:

Theorem 2 *If there is a function in* NP *which is infinitely often balanced and* $(1-1/\mathrm{poly}(n))$*-hard for circuits of polynomial size, then there is a function in* NP *which is infinitely often* $(\frac{1}{2}+n^{-1/2+\epsilon})$*-hard for circuits of polynomial size. (Here "infinitely often" simply means for infinitely many input lengths* n*.)*

The tribes function is nearly optimal given our techniques; getting hardness down to, say, $\frac{1}{2}+1/n$, would require a significant departure, if indeed this is even possible (see the discussion at the end of Section 7).

Finally, let us remark that the assumption in Theorem 2 that the initial hard function is balanced may be removed at the expense of a small loss in final hardness:

Theorem 3 *If there is a function in* NP *which is infinitely often* $(1-1/\mathrm{poly}(n))$*-hard for circuits of polynomial size, then there is a function in* NP *which is infinitely often* $(\frac{1}{2}+n^{-1/3+\epsilon})$*-hard for circuits of polynomial size.*

1.1 Organization of the paper

In Section 2 we give an intuitive discussion explaining why Answer 1 is the correct hardness bound; then in Section 3 we give the computational proof of Theorem 1. Section 4 introduces another property of boolean functions called *noise stability*, which we relate to expected bias. In Section 5 we study the expected bias of the majority function and conclude that it is not a particularly good hardness amplifier. Section 6 contains a estimation of the noise stability of the recursive majority of 3 function which allows us to prove a theorem which reduces $1-1/\mathrm{poly}(n)$ hardness to $1/2+o(1)$ hardness under the assumption that the initial hard function is balanced. Section 7 investigates the tribes function and completes the proof of Theorem 2, reducing hardness to $1/2+n^{-1/2+\epsilon}$. At the end of this section, we explain why we have essentially exhausted the limits of Theorem 1's applicability to monotone functions. Finally, Section 8, is devoted to proving Theorem 3.

Appendix A contains a technical proof omitted from the proof of Theorem 1 in Section 3. Appendix B gives constructions showing that Theorem 1 is essentially tight.

1.2 Related work

An important earlier work on which we rely is the "hard-core set" paper of Impagliazzo [Imp95]. Impagliazzo's paper shows that for any function f which is $(1-\delta)$-hard, there is a $(2-o(1))\delta$-fraction of inputs on which f is extremely hard — essentially unpredictable; whereas, off this hard-core set, f may be very easy. With this result in hand, it's possible to formalize the information-theoretic

reasoning behind hardness amplification. For example, Impagliazzo used his result to give a very simple and intuitive proof of the Yao XOR Lemma. Similarly, we prove Theorem 1 by applying Impagliazzo's construction, and then formalizing the intuition outlined in Section 2.

As we will see, the expected bias of a function is closely related to another one of its intrinsic properties, namely its sensitivity to noise in the input. Quantities related to these have previously been studied in [BKS99] and [BJT99]. The former is an extensive work by Benjamini, Kalai, and Schramm on the noise sensitivity and stability of boolean functions. It relates noise sensitivity to Fourier coefficients, proving a formula similar to our Proposition 9. It also gives some mention to the recursive majority of 3 function and tribes function which we treat in more detail in Section 6.

The second work mentioned is a learning theory paper by Bshouty, Jackson, and Tamon, which has a similar spirit to the present paper. They define a quantity called "noisy distance", which is much like expected bias or noise stability, and they give lower bounds for learning under attribute noise in terms of this quantity. They also prove a result much like our Proposition 7, and their main theorem is proved via a hybrid argument with some of the same flavor as Theorem 1's proof.

Subsequent to the present work, Elchanan Mossel and the author gave a fairly complete study [MO02] of the noise sensitivity of monotone functions.

Fourier analysis often greatly aids the study of structural properties of boolean functions, and indeed spectral considerations lie hidden below much of the work in this paper, arising explicitly only in the analysis of the noise stability of the tribes function. Relevant work on the Fourier analysis of boolean functions — especially monotone ones — appears in [KKL88, BT96, Man98]. The two monotone functions we use to amplify hardness were both introduced in [BL90], as examples of functions in which individual variables have small influence.

The Yao XOR Lemma, and direct product lemmas in general, are studied in [GNW95] and [Sha01] respectively. The constructions of [Sha01] are used heavily in Appendix B. Other papers which describe what boolean functions can or cannot do based on spectral properties include [LMN93, GR00].

2 Intuition for Theorem 1

Roughly speaking, Yao's XOR lemma says that if f is $(1-\delta)$-hard, then $f \oplus f \oplus \cdots \oplus f$ (k times) is $(\frac{1}{2} + \frac{1}{2}(1-\delta)^k)$-hard. The intuition is that in order to calculate $f(x_1) \oplus \cdots \oplus f(x_k)$, one must correctly calculate each $f(x_i)$ — if any are missed, one might as well guess at the final answer. Since f is $(1-\delta)$-hard and $x_1, \ldots, x_k$ are independent, we should only get advantage $(1-\delta)^k$, hence hardness $\frac{1}{2} + \frac{1}{2}(1-\delta)^k$.

To see the intuition behind Answer 1 we have to think more carefully about the issue of "knowing" whether or not $f(x_i)$ was calculated correctly.

Suppose that $x_1, \ldots, x_k$ are picked at random, and our task is to compute $(g \otimes f)(x_1, \ldots x_k) = g(f(x_1), \ldots, f(x_k))$, where f is a balanced function which is $(1-\delta)$-hard. We would like to abstract away the hard function f and consider instead the problem of calculating g with imperfect information. Let's call $y_i = f(x_i)$, $i = 1 \ldots k$ the "true" inputs to g. Since f is balanced and the x_i's are independent, it's as if the true inputs were also chosen independently and uniformly from $\{0,1\}^k$. Let's say that the hardness of f obscures our view of what the true inputs are, and we see only "corrupted" inputs $z_1, \ldots, z_k$, which match the true inputs only with probability $1-\delta$. Our goal is to guess $g(y_1, \ldots, y_k)$ as best we can given that we see only $z_1, \ldots, z_k$.

One way to model the hardness of f would simply be to say that $z_i = y_i$ independently with probability $1-\delta$. In this case, our best strategy would be a maximum likelihood one, involving taking a weighted majority of the values of g, with the most weight being given to points near

$(z_1, \ldots, z_k)$ in Hamming distance. However, it's possible we might actually have more information than this model allows — one could imagine that for certain recognizable inputs x_i it is very easy to calculate $f(x_i)$, and we can be sure our z_i is correct. Indeed, it's possible that for a large fraction of the inputs, we not only know the answer, but we *know* we know. Taking this to its extreme, we come to the most most helpful way in which the true inputs could be obscured:

> For each i, with probability $1 - 2\delta$, z_i is the correct value of y_i and furthermore *we know that this corrupted input is correct*; with probability 2δ, z_i is set to a random bit and *we know that this corrupted input is completely uncorrelated to* y_i.

Definition 6 *We call the above situation the* Hard-Core *scenario (cf. [Imp95]).*

Notice that in the Hard-Core scenario, each bit z_i is still correct with probability $1 - \delta$. To see that this arrangement is more useful to us than the first one, simply note that we can simulate the information we had in the first scenario by ignoring the extra knowledge about whether each z_i is correct or random.

The Hard-Core scenario in some sense gives us the most information — i.e., it models f as being $(1 - \delta)$-hard in the easiest possible way. The optimal strategy for guessing g in this scenario is clear. We look at the restricted version of the function g gotten when the z_i's which are known to be correct are plugged in. Since the remaining inputs are completely unknown, but also completely random, we should guess according to the bias of the restricted function. I.e., if the function is biased towards 0, guess 0; if it is biased towards 1, guess 1.

The probability that this strategy is correct is exactly the expected bias of g_ρ over random restrictions ρ with $\star$-probability 2δ — i.e., $\mathrm{ExpBias}_{2\delta}(g)$. This is essentially the hardness we prove for $g \otimes f$ in Theorem 1.

3 The hardness theorem

In this section we give the computational proof of Theorem 1, modeled after the intuitive discussion of the previous section. Our main tool is the hard-core set theorem of Impagliazzo [Imp95]. Roughly speaking, this theorem says that in the computational context of circuits, every $(1-\delta)$-hard function conforms to the Hard-Core scenario described in Section 2. That is, on a 2δ-fraction of its inputs the function is nearly impossible for small circuits, whereas on the remainder of its inputs it may be very easy.

We record here as a lemma the exact statement of Impagliazzo's theorem that we need:

Lemma 4 *Let f be $(1 - \delta)$-hard for size s. Then for every constant $r > 0$, f has a "hard-core" $S \subseteq \{0,1\}^n$ of size at least $(2 - r)\delta 2^n$ and at most $2\delta 2^n$ with the following property: on S, f is balanced and $(1/2 + \epsilon)$-hard for size $\Omega(\epsilon^2/\log(1/\delta)\, s)$.*

Proof: Combining results from [Imp95] and [KS99], there must exist a set $S' \subseteq \{0,1\}^n$ of size at least $(2 - r)\delta 2^n$ and at most $(2 - r/2)\delta 2^n$ on which f is $(1/2 + \epsilon/2)$-hard for size $\Omega(\epsilon^2/\log(1/\delta)\, s)$. We would like to add a small number of strings to S' to make it balanced. Suppose without loss of generality that f is biased towards 1 on S'. Since f is $(1/2 + \epsilon/2)$-hard on S', it is 1 on at most $(1/2 + \epsilon/2)|S'|$ strings in S'. Since $|S'| \leq (2 - r/2)\delta 2^n$ and we may assume without loss of generality that $\epsilon < r/4$, the total number of 1's f has in S' is at most $(1/2 + \epsilon/2)(2 - 2\epsilon)\delta 2^n \leq \delta 2^n$.

On the other hand, since f is $(1-\delta)$-hard on $\{0,1\}^n$, f must take on the value 0 for at least $\delta 2^n$ strings in $\{0,1\}^n$. It follows that there is a superset $S' \subseteq S \subseteq \{0,1\}^n$ of S' on which f is balanced. This set has size at most $2\delta 2^n$.

It remains to show that f is $(1/2+\epsilon)$-hard on S. We know that no circuit (of size $\Omega(\epsilon^2/\log(1/\delta)\ s)$) gets f right on S' with probability more than $1/2+\epsilon/2$, and no circuit gets f right on $S \setminus S'$ with probability more than 1. We know that $|S \setminus S'| = \text{bias}(f|_{S'})|S'| - (1-\text{bias}(f|_{S'}))|S'| = (2\text{bias}(f|_{S'})-1)|S'| \leq \epsilon|S'|$. It follows that no circuit gets f right on S with probability more than:

$$\begin{aligned}
(1/2+\epsilon/2)\frac{|S'|}{|S|} + \frac{|S \setminus S'|}{|S|} &= (1/2+\epsilon/2)\frac{|S| - |S \setminus S'|}{|S|} + \frac{|S \setminus S'|}{|S|} \\
&= 1/2 + \epsilon/2 + (1/2-\epsilon/2)\frac{|S \setminus S'|}{|S|} \\
&\leq 1/2 + \epsilon/2 + (1/2-\epsilon/2)\epsilon \\
&\leq 1/2 + \epsilon,
\end{aligned}$$

as needed. □

With Impagliazzo's theorem formalizing the reason for which f is hard, there is only one more ingredient we need to complete the proof along the lines outlined in Section 2. We need to show that if all of a function's inputs look completely random then it is impossible to guess its value with probability better than its bias.

Lemma 5 *Let $\mathcal{B}_n$ denote the n-dimensional Hamming cube, and suppose $h : \mathcal{B}_n \to \{0,1\}$ and $p : \mathcal{B}_n \to [0,1]$. Further suppose that*

$$2^{-n}\left[\sum_{x \in h^{-1}(0)} p(x) + \sum_{x \in h^{-1}(1)} (1-p(x))\right] \qquad (*)$$

is at least $\text{bias}(h) + \epsilon$. *Then there exists an edge (x,y) in $\mathcal{B}_n$ such that $|p(x) - p(y)| \geq \Omega(\epsilon/\sqrt{n})$.*

Proof: The idea here is that h is the function whose value on $\{0,1\}^n$ we are trying to guess, and $p(x)$ represents the probability with which we guess 0 when the input is x. Our success probability is given by $(*)$, and the lemma says that if we are doing strictly better than $\text{bias}(h)$ then we must be doing at least *some* distinguishing between adjacent points in the cube; i.e., the points in the cube cannot look completely indistinguishable to us.

Getting the lower bound $\Omega(\epsilon/\sqrt{n})$ is a little tricky, so we defer the proof to Appendix A. For now we just prove a lower bound of ϵ/n. Note that using this weaker lower bound in the proof of Theorem 1 makes no qualitative difference, and we could still prove Theorem 2 just as well.

Without loss of generality, suppose h is biased towards 0, and let $b = \text{bias}(h) = \Pr[h=0] \geq 1/2$. By way of contradiction, assume $|p(x) - p(y)| < \epsilon/n$ for all edges $(x,y) \in \mathcal{B}_n$. Let M and m be the maximum and minimum values of p on $\mathcal{B}_n$. Since any of pair of points in the cube is at Hamming distance at most n, it follows that $M - m < \epsilon$. Now $(*)$ would be maximized if $p(x) = M$ for all

$x \in h^{-1}(0)$ and $p(x) = m$ for all $x \in h^{-1}(1)$. Hence:

$$\begin{aligned}
(*) &\leq bM + (1-b)(1-m) \\
&= (M+m-1)b + 1 - m \\
&\leq (2M-1)b + 1 - m + M - M \\
&< (2M-1)b + 1 - M + \epsilon \\
&= (1-M) - (2-2M)b + b + \epsilon \\
&\leq b + \epsilon,
\end{aligned}$$

since $b \geq 1/2$. This contradiction completes the proof. □

Now we prove the hardness theorem. We will actually prove a slightly stronger statement than the one given in Section 1 — for techincal reasons we want the theorem to hold under the assumption that the hard function f is only nearly balanced, not necessarily exactly balanced.

Theorem 1 *Let $f : \{0,1\}^n \to \{0,1\}$ be a function which is $(1-\delta)$-hard for size s. Let $g : \{0,1\}^k \to \{0,1\}$ be any function. Let $\epsilon > 0$ be any parameter, and assume that* $\mathrm{bias}(f) \leq 1/2 + (1-2\delta)\epsilon/4k$. *Then for every $r > 0$, $g \otimes f$ is $(\mathrm{ExpBias}_{(2-r)\delta}(g) + \epsilon)$-hard for circuits of size $s' = \Omega(\frac{\epsilon^2/\log(1/\delta)}{k} s)$.*

Proof: Let S be the hard-core given by Lemma 4, using parameters δ, r, and $\epsilon' = (c/8)\epsilon/\sqrt{k}$, where c is the constant hidden in the $\Omega(\cdot)$ of Lemma 5. Then f is $(1/2+\epsilon')$-hard on S for circuits of size s'. Furthermore, we have $\mathrm{bias}(f|_S) = 1/2$, and because $|S| \leq 2\delta 2^n$ and $\mathrm{bias}(f) \leq 1/2 + (1-2\delta)\epsilon/4k$, we get $\mathrm{bias}(f|_{S^c}) \leq 1/2 + \epsilon/4k$.

Let $\eta = |S|/2^n \geq (2-r)\delta$, and let $E = \mathrm{ExpBias}_\eta(g)$. It's easy to see that $\mathrm{ExpBias}_\delta(h)$ is a nonincreasing function of δ for any h — simply note that in the guessing game of Section 2, the more information the optimal strategy has the better. Hence $E \leq \mathrm{ExpBias}_{(2-r)\delta}(g)$.

Suppose by way of contradiction that C is a circuit of size s' computing $g \otimes f$ on an $(\mathrm{ExpBias}_{(2-r)\delta}(g) + \epsilon) \geq E + \epsilon$ fraction of the inputs in $(\{0,1\}^n)^k$. For a given restriction $\rho \in P_\eta^k$, let's say that an input $(x_1, \ldots, x_k)$ *matches* ρ when for all i we have $x_i \in S \Rightarrow \rho(i) = \star$ and $x_i \in S^c \Rightarrow \rho(i) = f(x_i)$. Note that the probability an input matches ρ is very nearly ρ's natural probability under P_η^k. In particular: the probability that $x_i \in S$ is exactly η, which is the correct $\star$ probability; the probability that $f(x_i) = 1$ given that $x_i \in S^c$ is at most $1/2 + \epsilon/4k$ because $\mathrm{bias}(f|_{S^c}) \leq 1/2 + \epsilon/4k$; and, a similar statement holds for $f(x_i) = 0$. Hence $\Pr[(x_1, \ldots, x_k) \text{ matches } \rho] \leq (1/2 + \epsilon/4k)^k/(1/2)^k \Pr[\rho] \leq (1 + (2/3)\epsilon)\Pr[\rho]$ for every ρ.

Let c_ρ be the probability that C correctly computes $(g \otimes f)(x_1, \ldots, x_k)$ conditioned on the event that $(x_1, \ldots, x_k)$ matches ρ. We have:

$$\begin{aligned}
& & \Pr[C \text{ correct}] &\geq E + \epsilon \\
&\Rightarrow & \textstyle\sum_\rho \Pr[(x_1, \ldots, x_k) \text{ matches } \rho] c_\rho &\geq \textstyle\sum_\rho \Pr[\rho]\mathrm{bias}(g_\rho) + \epsilon \\
&\Rightarrow & \textstyle\sum_\rho (1 + (2/3)\epsilon)\Pr[\rho] c_\rho &\geq \textstyle\sum_\rho \Pr[\rho]\mathrm{bias}(g_\rho) + \epsilon,
\end{aligned}$$

and it follows that there must exists a restriction ρ such that $c_\rho \geq \mathrm{bias}(g_\rho) + \epsilon/4$.

By reordering the k length-n inputs to C we may assume ρ is of the form $(\underbrace{\star, \ldots, \star}_{k' \text{ times}}, b_{k'+1}, \ldots, b_k)$ for some $k' < k$, $b_i \in \{0,1\}$. An averaging argument tells us there exist particular $x^*_{k'+1}, \ldots, x^*_k \in \{0,1\}^n$ with $f(x^*_i) = b_i$ such that C correctly computes $g_\rho \otimes f$ with probability at least c_ρ when the first k' inputs are drawn independently and uniformly from S, and the last $k - k'$ inputs are hardwired to the x^*'s.

Let C' be the size s' circuit given by hardwiring in the x^*'s. So C' is a circuit taking k' strings in $\{0,1\}^n$, and it correctly computes g_ρ with probability at least c_ρ when its inputs $x_1, \ldots, x_{k'}$ are drawn independently and uniformly from S. From now on, whenever we speak of probabilities with respect to C' we mean over uniformly random inputs drawn from S (not all of $\{0,1\}^n$).

For each $(y_1, \ldots, y_{k'}) \in \{0,1\}^{k'}$, let $p(y)$ be the probability that $C'(x_1, \ldots, x_{k'}) = 0$ conditioned on the event that $y_i = f(x_i)$ for all $i = 1 \ldots k'$. Since f is balanced on S, each of these events is equally likely. It follows that the correctness probability of C' is exactly:

$$2^{-k'} \left[\sum_{y \in g_\rho^{-1}(0)} p(y) + \sum_{y \in g_\rho^{-1}(1)} (1 - p(y)) \right].$$

Since this quantity is at least $c_\rho \geq \text{bias}(g_\rho) + \epsilon/4$, Lemma 5 tells us that there is an edge (z, z') in $\mathcal{B}_{k'}$ such that $|p(z) - p(z')| \geq c(\epsilon/4)/\sqrt{k'} \geq (c/4)\epsilon/\sqrt{k} = 2\epsilon'$.

Reorder inputs again so that we may assume $z = 0u$, $z' = 1u$ for some string $u \in \{0,1\}^{k'-1}$. Again, an averaging argument tells us that there exist $x_2^*, \ldots x_{k'}^*$ with $(f(x_2^*), \ldots, f(x_{k'}^*)) = u$ such that

$$\left| \Pr_{x_1 \in (f|_S)^{-1}(0)} [C'(x_1, x_2^*, \ldots x_{k'}^*) = 0] \; - \Pr_{x_1 \in (f|_S)^{-1}(1)} [C'(x_1, x_2^*, \ldots x_{k'}^*) = 0] \right| \geq 2\epsilon'.$$

Now let C'' be the size s' circuit given by hardwiring in the new x^*'s. Then:

$$\left| \Pr_{x \in (f|_S)^{-1}(0)} [C''(x) = 0] \; - \Pr_{x \in (f|_S)^{-1}(1)} [C''(x) = 0] \right| \geq 2\epsilon',$$

and so we have a circuit of size s' which when given one random input from S can distinguish the cases $f(x_1) = 0$ and $f(x_1) = 1$ with advantage $2\epsilon'$. This contradicts the fact that f is $(\frac{1}{2} + \epsilon')$-hard for size s'. □

In Appendix B we give unconditional constructions showing that this theorem is nearly tight.

4 Noise stability

We now know that the expected bias of g essentially characterizes the hardness of a composite function $g \otimes f$. Unfortunately, $\text{ExpBias}_\delta(g)$ is often difficult to calculate, even for fairly simple functions g. In this section we introduce another intrinsic property of boolean functions called *noise stability* which turns out to be a very good estimator for expected bias and is in general easier to calculate. Noise stability has been studied in a number of other works; e.g., [BKS99, BJT99, MO02].

Let's begin with some notation. Because any binary function can be guessed with probability at least $1/2$, we often deal with probabilities in the range $[1/2, 1]$. Sometimes it is more natural to work with the amount these quantities are in excess of $1/2$, scaled to the range $[0, 1]$. To that end, we introduce the following notational quirk:

Definition 7 *If Q is any quantity in the range $[1/2, 1]$, Q^* will denote the quantity $2(Q - 1/2)$, which is in the range $[0, 1]$.*

We give bias* a special name:

Definition 8 *The* advantage *of a boolean function h is* $\text{adv}(h) = \text{bias}(h)^*$.

Let h be any boolean function on n bits. The advantage of h is related to the probability that h agrees on two random values. The following fact is easy to prove:

Proposition 6 *If x and y are selected independently and uniformly from $\{0,1\}^n$, the probability that $h(x) = h(y)$ is $\frac{1}{2} + \frac{1}{2}\mathrm{adv}(h)^2$.*

Instead of picking x and y at random, we might first pick x at random and then make y a random perturbation of x with noise δ.

Definition 9 *If $x \in \{0,1\}^n$, define $N_\delta(x)$ to be a random variable in $\{0,1\}^n$ given by flipping each bit of x independently with probability δ.*

Now we can define the noise stability of a boolean function h.

Definition 10 *The* noise stability of h at δ *is:*

$$\mathrm{NoiseStab}_\delta(h) = \Pr[h(x) = h(N_\delta(x))],$$

where the probability is taken over the choice of x and the noise.

We can see the connection between noise stability and expected bias by recalling the intuitive discussion from Section 2. Suppose we are in the Hard-Core scenario, in which with probability $1-2\delta$ we know that z_i is the correct answer $f(x_i)$, and with probability 2δ we know that z_i is a completely random bit. Then $\mathrm{NoiseStab}_\delta(g)$ is exactly the success probability of the suboptimal, naive strategy of just guessing $g(z_1, \ldots, z_k)$ for the value of $g(f(x_1), \ldots, f(x_k))$. Since the optimal strategy had a success probability of $\mathrm{ExpBias}_{2\delta}(g)$, we immediately conclude that $\mathrm{NoiseStab}_\delta(h) \leq \mathrm{ExpBias}_{2\delta}(h)$ for all functions h.

Perhaps surprisingly, the naive guessing strategy is not so bad. The following result shows that if we look at the advantage over $1/2$ of the naive strategy and the optimal strategy, the naive strategy is worse by at most a square root.

Proposition 7 $\mathrm{NoiseStab}_\delta(h)^* \leq \mathrm{ExpBias}_{2\delta}(h)^* \leq \sqrt{\mathrm{NoiseStab}_\delta(h)^*}$.

Proof: Given a $\rho \in P_\delta^n$, write $\mathrm{stars}(\rho)$ for the set of coordinates to which ρ assigns $\star$. By linearity of expectation, $\mathrm{ExpBias}_{2\delta}(h)^* = \mathbf{E}_{\rho \in P_{2\delta}^n}[\mathrm{adv}(h_\rho)]$. On the other hand,

$$\begin{aligned}
\mathrm{NoiseStab}_\delta(h) &= \Pr_{x,\mathrm{noise}}[h(x) = h(N_\delta(x))] \\
&= \Pr_{\substack{\rho \in P_{2\delta}^n \\ y,z \in \{0,1\}^{\mathrm{stars}(\rho)}}}[h_\rho(y) = h_\rho(z)] \\
&= \mathbf{E}_\rho[\Pr_{y,z}[h_\rho(y) = h_\rho(z)] \\
&= \mathbf{E}_\rho\left[\frac{1}{2} + \frac{1}{2}\mathrm{adv}(h_\rho)^2\right] \qquad \text{(by Proposition 6)}
\end{aligned}$$

and so $\mathrm{NoiseStab}_\delta(h)^* = \mathbf{E}_\rho[\mathrm{adv}(h_\rho)^2]$. Since $\mathrm{adv} \in [0,1]$, we get the left inequality since $\mathrm{adv}^2 \leq \mathrm{adv}$, and we get the right inequality by Cauchy-Schwarz. □

Proposition 7 is useful because it tells us that $\mathrm{NoiseStab}_\delta(h)$ is a good estimator for $\mathrm{ExpBias}_{2\delta}(h)$; the two quantities are qualitatively similar in that if one is $1 - o(1)$ then so is the other, and similarly for $1 - \Omega(1)$ and $1/2 + o(1)$. It's also satisfying to confirm that functions which are highly

noise-unstable make good hardness amplifiers — this is roughly the property of XOR to which one appeals in intuition for the XOR Lemma.

We end this section with two tools which help in calculating noise stability. The first is a simple but useful observation which follows immediately from the definitions:

Proposition 8 *If h is a balanced boolean function, and g is any boolean function, then* $\mathrm{NoiseStab}_\delta(g \otimes h) = \mathrm{NoiseStab}_{1-\mathrm{NoiseStab}_\delta(h)}(g)$.

The second tool is a formula for noise stability in terms of Fourier coefficients; similar formulas appear in [BKS99, BJT99]. Let us set up the usual Fourier machinery. We begin with an important notational convention: Whenever we discuss Fourier coefficients, instead of *true* being represented by 1 and *false* by 0, we will use -1 and $+1$, respectively.

Let $h : \{+1,-1\}^n \to \{+1,-1\}$ be any boolean function. Then h has a unique representation as a multilinear polynomial in $x_1, \ldots, x_n$ of total degree at most n. The coefficients of this polynomial are the Fourier coefficients of h. Let x_S denote the monomial $\prod_{i \in S} x_i$, where $S \subseteq [n]$. Then the Fourier coefficient corresponding to this monomial is denoted $\widehat{h}(S)$. Note that $\widehat{h}(\emptyset) = \mathbf{E}[h]$.

The formula for noise stability is as follows:

Proposition 9 *Let $h : \{+1,-1\}^n \to \{+1,-1\}$. Then:*

$$\mathrm{NoiseStab}_\delta(h)^* = \sum_{S \subseteq [n]} (1-2\delta)^{|S|} \widehat{h}(S)^2.$$

Proof: Let $x \in \{+1,-1\}^n$ be chosen uniformly at random, and let $y = N_\delta(x)$. Then:

$$\begin{aligned}
\mathrm{NoiseStab}_\delta(h) &= \Pr[h(x) = h(y)] \\
&= \mathbf{E}[\mathbf{1}_{h(x)=h(y)}] \\
&= \mathbf{E}[1 - (h(x) - h(y))^2/4] \quad (\text{since } h \in \{+1,-1\}) \\
&= 1 - \tfrac{1}{4}\mathbf{E}[h(x)^2] - \tfrac{1}{4}\mathbf{E}[h(y)^2] + \tfrac{1}{2}\mathbf{E}[h(x)h(y)] \\
&= \tfrac{1}{2} + \tfrac{1}{2}\mathbf{E}[h(x)h(y)] \qquad (\text{since } h^2 \equiv 1);
\end{aligned}$$

hence $\mathrm{NoiseStab}_\delta(h)^* = \mathbf{E}[h(x)h(y)]$. But

$$\begin{aligned}
\mathbf{E}[h(x)h(y)] &= \mathbf{E}[(\sum_S \widehat{h}(S)x_S)(\sum_T \widehat{h}(T)y_T)] \\
&= \sum_{S,T} \widehat{h}(S)\widehat{h}(T)\,\mathbf{E}[x_S y_T].
\end{aligned}$$

If $S \neq T$, it is fairly easy to see that $\mathbf{E}[x_S y_T] = 0$: Let $i \in S \Delta T$; say $i \in T \setminus S$. Now y_i is equally likely to be $+1$ or -1, and hence $\mathbf{E}[x_S y_T] = \frac{1}{2}\mathbf{E}[x_S y_T | y_i = +1] + \frac{1}{2}\mathbf{E}[x_S y_T | y_i = -1] = \frac{1}{2}\mathbf{E}[x_S y_{T\setminus\{i\}}] - \frac{1}{2}\mathbf{E}[x_S y_{T\setminus\{i\}}] = 0$.

It remains to prove that $\mathbf{E}[x_S y_S] = (1-2\delta)^{|S|}$. Since x_S and y_S are both either $+1$ or -1, $\mathbf{E}[x_S y_T] = \Pr[x_S = y_S] - \Pr[x_S \neq y_S] = -1 + 2\Pr[x_S = y_S]$. This last probability is exactly the probability that, if we flip $|S|$ coins with probability δ for heads, then we get an even number of heads. A trivial induction shows that this is indeed $\frac{1}{2} + \frac{1}{2}(1-2\delta)^{|S|}$; another way to see it is to write $k = |S|$, and note that the probability is:

$$\begin{aligned}
&(1-\delta)^k + \binom{k}{2}(1-\delta)^2\delta^2 + \binom{k}{4}(1-\delta)^4\delta^4 + \ldots \\
= \;& ([(1-\delta)+\delta]^k + [(1-\delta)-\delta]^k)/2 = \frac{1}{2} + \frac{1}{2}(1-2\delta)^k,
\end{aligned}$$

as claimed. □

5 Majority

This section is devoted to one of the simplest monotone functions, the majority function. It turns out that the majority function is not a very good hardness amplifier, but it is nevertheless interesting to study its expected bias. Let MAJ_n denote the majority function on n bits; for simplicity we assume n to be odd.

Proposition 10 *For any* $\delta \in [0,1]$, $\mathrm{ExpBias}_\delta(\mathsf{MAJ}_n) \to \frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2\delta)$ *as* $n \to \infty$. *In particular:*

$$\left|\mathrm{ExpBias}_\delta(\mathsf{MAJ}_n) - \left(\frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2\delta)\right)\right| \leq O(1/\sqrt{\delta(1-\delta)n}).$$

Proof: Suppose $\rho \in P_\delta^n$ is chosen at random. Then $(\mathsf{MAJ}_n)_\rho$ is biased towards the majority value of the non-star bits of ρ. Indeed, $\mathrm{bias}((\mathsf{MAJ}_n)_\rho)$ is exactly equal to the probability that, if one proceeds to set the starred bits of ρ at random, the majority is preserved. (This is ambiguous in the case that the non-star bits of ρ split evenly; we shall see that we needn't worry about this case.)

Let us model this random experiment as follows: Suppose we consider each bit position independently and in sequence, 1, 2, ... , n. For each position i, we first choose whether i will be a "star position" (probability δ) or a "non-star position" (probability $1-\delta$). Either way, we also select a random bit associated to position i (equally likely 0 or 1). We wish to calculate the probability that the majority of all the bits agrees with the majority of the non-star bits.

Our experiment corresponds to a random walk on the lattice points of $\mathbf{R}^2$ in a natural way: Let star positions correspond to vertical steps and non-star positions correspond to horizontal steps; further, let right and up steps correspond to choosing the bit 0 and left and down steps correspond to choosing 1. Now we have an n-step random walk starting at the origin, where each step is: $(+1,0)$ with probability $(1-\delta)/2$, $(-1,0)$ with probability $(1-\delta)/2$, $(0,+1)$ with probability $\delta/2$, or $(0,-1)$ with probability $\delta/2$. Let (X,Y) denote the final position of the random walk. Then the majority of all the bits agrees with the majority of the non-star bits if and only if: $X+Y \geq 0$ and $X \geq 0$; or, $X+Y \leq 0$ and $X \leq 0$. (Again, the boundary of this region falls into an ambiguous case, but this will be unimportant.) Call this region R; R is shown shaded on the left of Figure 1.

Let us write $X_1, \ldots, X_n \in \{-1,0,+1\}$ for the x-components of the random walk's n steps, and similarly $Y_1, \ldots, Y_n$ for the y-components. We have $X = X_1 + \cdots + X_n$, $Y = Y_1 + \cdots + Y_n$, and we wish to study the distribution of (X,Y). Note that the mean of one step in the random walk (X_i, Y_i) is $(0,0)$ and the covariance matrix for one step is $\Sigma = \mathbf{E}\begin{pmatrix} X_i^2 & X_iY_i \\ Y_iX_i & Y_i^2 \end{pmatrix} = \begin{pmatrix} 1-\delta & 0 \\ 0 & \delta \end{pmatrix}$. By the multidimensional Central Limit Theorem, the distribution function of $(X/\sqrt{n}, Y/\sqrt{n})$ converges to that of the 2-dimensional Gaussian with covariance Σ; namely, the distribution with density function:

$$\phi_\Sigma(x,y) = \frac{1}{2\pi\sqrt{\delta(1-\delta)}} \exp\left(-\left(\frac{x^2}{1-\delta} + \frac{y^2}{\delta}\right)/2\right).$$

Sazonov gave Berry-Esséen-type bounds for the rate of convergence in the multidimensional Central Limit Theorem; in particular, [Saz81, p. 10 item 6] shows that for any convex set S in $\mathbf{R}^2$, $|\Pr[(X/\sqrt{n}, Y/\sqrt{n}) \in S] - \Phi_\Sigma(S)| \leq O(1)(1/\sqrt{\delta} + 1/\sqrt{1-\delta})n^{-1/2} = O(1/\sqrt{\delta(1-\delta)n})$, where Φ_Σ denotes the distribution function of the Gaussian distribution mentioned above. Since R is the union of two convex sets, and $(X/\sqrt{n}, Y/\sqrt{n}) \in R$ iff $(X,Y) \in R$ by definition of R, the proof

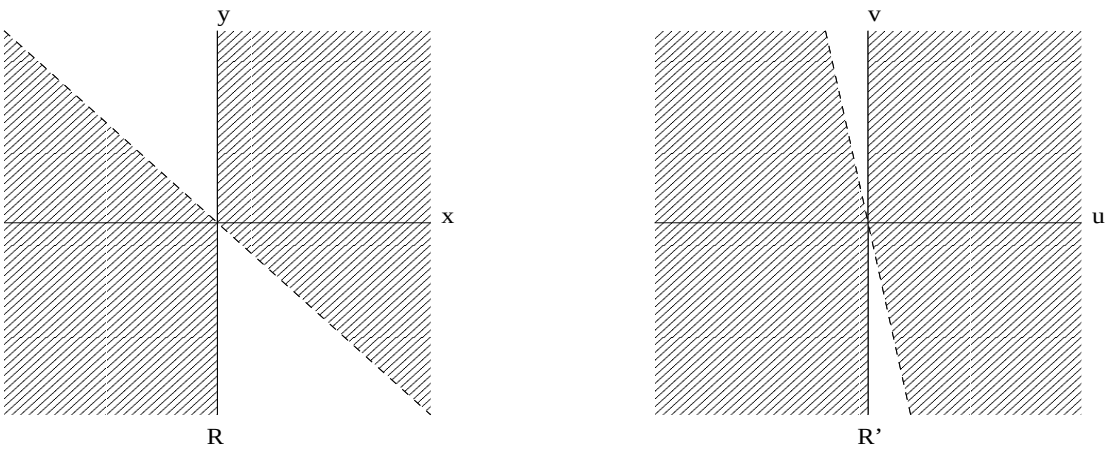

Figure 1: The regions R and R'.

will be complete once we show that $\Phi_\Sigma(R) = \frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2\delta)$. (It is now apparent that the definition of R's boundary is irrelevant since it gets 0 weight under the continuous distribution ϕ_Σ.)

This calculation is easy. Make the change of variables $u = x/\sqrt{1-\delta}$, $v = y/\sqrt{\delta}$. Then ϕ_Σ becomes the standard 2-dimensional normal distribution, $\phi(u,v) = \frac{1}{2\pi}\exp(-(u^2+v^2)/2)$, and R is transformed into $R' = \{u \geq 0, \sqrt{1-\delta}\,u + \sqrt{\delta}\,v \geq 0\} \cup \{u \leq 0, \sqrt{1-\delta}\,u + \sqrt{\delta}\,v \leq 0\}$, pictured on the right in Figure 1. Now ϕ is evidently radially symmetric, and R' consists of exactly two radial slices each of angle $\pi/2 + \tan^{-1}\sqrt{(1-\delta)/\delta}$. Therefore the weight of R' under Φ is precisely $(1/2\pi)\cdot 2(\pi/2 + \tan^{-1}\sqrt{(1-\delta)/\delta}) = 1/2 + (1/\pi)\tan^{-1}\sqrt{(1-\delta)/\delta} = \frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2\delta)$, by a trigonometric identity. □

It is not surprising that arcsin should arise in this context; c.f. the arc sine laws of Feller [Fel68].

Roughly speaking, Theorem 1 now tells us that if f is balanced and $(1-\delta)$-hard, then $\mathsf{MAJ} \otimes f$ has hardness about $\frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2(2\delta))$. The graph of $\frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2(2\delta))$ and $1-\delta$ versus δ is shown in Figure 2.

We can see then that when δ is small (i.e., f is "easy"), majority is a fair hardness amplifier; an approximation shows that it amplifies $1-\delta$ hardness to $1-\Theta(\sqrt{\delta})$ hardness. So long as $\delta < 1/4$, majority increases hardness; but, when $\delta > 1/4$, it actually *decreases* hardness! (It is easy to show that $\mathrm{ExpBias}_{1/2}(\mathsf{MAJ}_n) = 3/4$ for all odd n.) It's clear then that the majority function will not give us the good hardness amplification properties we desire.

6 Recursive majorities

It seems that recursive majorities might not be such a good idea — if one did the analysis simply by applying Theorem 1 with Proposition 10 multiple times, the hardness would converge and get "stuck" at 3/4. This would be giving too much away, however. Recall the intuitive discussion from Section 2; it has to assume that the function f is $(1-\delta)$-hard in the easiest possible way. But if f is already the majority of hard functions, it's in fact hard in a harder way. The key is to apply all of the majorities at the same time, taking g to be recursive-majority right from the start and applying Theorem 1 only once.

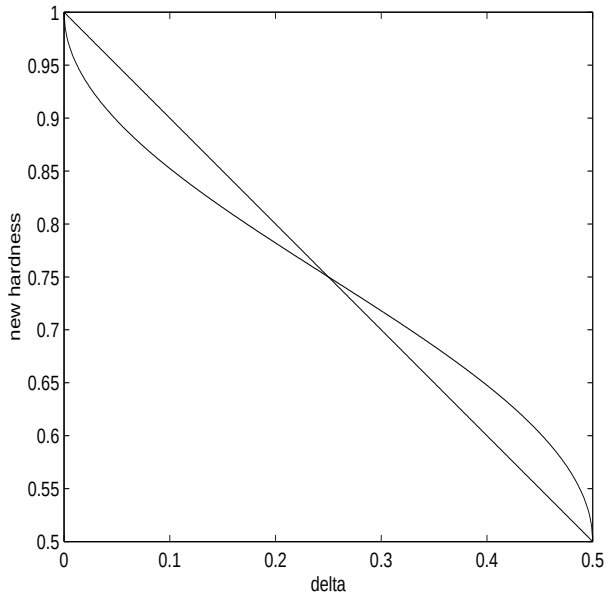

Figure 2: $\frac{3}{4} + \frac{1}{2\pi}\arcsin(1-2(2\delta))$ and $1-\delta$ as functions of δ.

Definition 11 *For each $\ell \geq 1$, $\mathsf{REC\text{-}MAJ\text{-}3}^{\ell} : \{0,1\}^{3^\ell} \to \{0,1\}$ is the function given by a depth-ℓ ternary tree of majority-of-3 gates, with the input bits at the leaves.*

We do not know how to calculate the expected bias of $\mathsf{REC\text{-}MAJ\text{-}3}^{\ell}$, but estimating its noise stability is straightforward. We give a crude estimate which suffices for our purposes:

Proposition 11 *For $\ell \geq \log_{1.1}(1/\delta)$, $\mathrm{NoiseStab}_\delta(\mathsf{REC\text{-}MAJ\text{-}3}^{\ell})^* \leq \delta^{-1.1}(3^\ell)^{-.15}$.*

Proof: If we write $\mathrm{NoiseInst}_\delta = 1 - \mathrm{NoiseStab}_\delta$, then Proposition 8 tells us that $\mathrm{NoiseInst}_\delta(g \otimes h) = \mathrm{NoiseInst}_{\mathrm{NoiseInst}_\delta(h)}(g)$ when h is balanced. MAJ_3 is balanced, and an explicit calculation gives $\mathrm{NoiseInst}_t(\mathsf{MAJ}_3) = (3/2)t - (3/2)t^2 + t^3$. Defining $p(t) = (3/2)t - (3/2)t^2 + t^3$, we conclude that $\mathrm{NoiseStab}_\delta(\mathsf{REC\text{-}MAJ\text{-}3}^{\ell}) = 1 - \underbrace{p(p(\cdots p(\delta)\cdots))}_{\ell \text{ times}}$, so $\mathrm{NoiseStab}_\delta(\mathsf{REC\text{-}MAJ\text{-}3}^{\ell})^* = 1 - 2p^{(\ell)}(\delta)$.

It remains to study the iteration of p. When t is sufficiently small, $p(t) \approx (3/2)t$. Very crudely, when $t < 1/4$, $p(t) \geq 1.18t$. It follows that $p^{(m)}(t) \geq (1.18)^m t$, so long as this quantity is at most $1/4$. Taking $m = \log_{1.18}(1/4\delta)$, we get $p^{(m)}(\delta) \geq (1/4)/1.18 > .21$.

Now write $t = 1/2 - \eta$. Then $p(t) = 1/2 - (3/4)\eta - \eta^3$. When η is very small, this is approximately $1/2 - (3/4)\eta$. Crudely again, when $\eta \leq 1/2 - .21 = .29$, $p(t) \geq 1/2 - .84\eta$. It follows that $p^{(m')}(.21) \geq 1/2 - (.84)^{m'}(.29)$ for any positive m', in particular for $m' = \ell - m$, which is positive because $\ell \geq \log_{1.1}(1/\delta) > m$.

Hence $p^{(\ell)}(\delta) \geq 1/2 - .29 \cdot (.84)^{m'}$, and so it remains to check that $.58 \cdot (.84)^{m'} \leq \delta^{-1.1}(3^\ell)^{-.15}$.

$$\begin{aligned} .58 \cdot (.84)^{m'} &= .58 \cdot (.84)^{\ell - \log_{1.18}(1/4\delta)} \\ &< (.84)^{-\log_{1.18}(1/4\delta)}(.84)^{\ell} \\ &= (4\delta)^{\log_{1.18}(.84)}(3^\ell)^{\log_3(.84)} \\ &< \delta^{-1.1}(3^\ell)^{-.15}, \end{aligned}$$

as needed. □

This result gives us a monotone balanced function which is very sensitive to a small amount of noise. Applying Proposition 7, we immediately get:

Proposition 12 *For $\ell \geq \log_{1.1}(1/\delta)$, $\mathrm{ExpBias}_{2\delta}(\text{REC-MAJ-3}^\ell)^* \leq \delta^{-.55}(3^\ell)^{-.075}$.*

We can now prove a $1 - 1/\mathrm{poly}(n) \to 1/2 + o(1)$ hardness amplification for NP under the assumption that the initial hard function in NP is balanced:

Theorem 13 *If there is a family of functions (f_n) in NP which is infinitely often balanced and $(1 - 1/\mathrm{poly}(n))$-hard for polynomial circuits, then there is a family of functions (h_m) in NP which is infinitely often balanced and $(1/2 + m^{-.07})$-hard for polynomial circuits.*

Proof: Suppose (f_n) is infinitely often balanced and $(1 - 1/n^c)$-hard for polynomial circuits. For a given n, let $k = n^C$, where C is a large constant dependent on c to be chosen later. Put $\ell = \lfloor \log_3 k \rfloor$ and view REC-MAJ-3^ℓ as a function on k bits by ignoring some inputs if necessary. Now define $h_m = \text{REC-MAJ-3}^\ell \otimes f_n$, having input length $m = kn = n^{C+1}$. Note that the family (h_m) is in NP since (f_n) is and since REC-MAJ-3^ℓ is monotone and in P. Also h_m is balanced whenever f_n is. It remains to check that h_m is indeed $(1/2 + m^{-.07})$-hard for polynomial circuits whenever n is such that f_n is hard.

Apply Theorem 1 with $r = 1$, $\epsilon = 1/n^C$, and $\delta = 1/n^c$. Then we conclude that h_m is $(\mathrm{ExpBias}_\delta(\text{REC-MAJ-3}^\ell) + \epsilon)$-hard for polynomial circuits. By Proposition 12, $\mathrm{ExpBias}_\delta(\text{REC-MAJ-3}^\ell) \leq 1/2 + (1/2)(\delta/2)^{-.55}(3^\ell)^{-.075}$, assuming $\ell \geq \log_{1.1}(2/\delta) = \log_{1.1}(2n^c)$; by taking C sufficiently large (compared to c) we can ensure this is true. But $(1/2)(\delta/2)^{-.55}(3^\ell)^{-.075} \leq (2n^c)^{.55}(n^C/3)^{-.075} + n^{-C} \leq n^{-.074C}$ if we take C large enough. And $n^{-.074C} \leq m^{-.07}$ if we take C large enough, since $m = n^{C+1}$. Thus h_m is $(1/2 + m^{-.07})$-hard for polynomial circuits, and the proof is complete. □

A more careful estimation in Proposition 11 would allow us to get a function which was $(1/2 + n^{-\alpha})$-hard for α approaching $\frac{1}{2}\log_3(4/3) \approx 0.13$. See [MO02] for a detailed analysis of the noise stability of recursive majorities.

7 The tribes function

The previous theorem got us from $1 - 1/\mathrm{poly}(n)$ hardness down to $1/2 + o(1)$ hardness. In this section we improve on the $o(1)$ term by moving to a different function, namely the "tribes" function of Ben-Or and Linial. This is a simple monotone read-once DNF, whose definition is as follows: For input length length k, the parameter b is set to a certain quantity a little less than $\log_2 k$. Then the k inputs are divided into k/b blocks of size b, and the function T_k is defined to be 1 if and only if at least one of the blocks consists entirely of 1's. I.e., $T_k(x_1, \ldots, x_k) = (x_1 \wedge \cdots \wedge x_b) \vee (x_{b+1} \wedge \cdots \wedge x_{2b}) \vee \cdots \vee (x_{k-b+1} \wedge \cdots \wedge x_k)$.

As in the previous section our technique is to estimate the noise stability of T_k. The calculation is more technical than before; we use the formula for noise stability in terms of Fourier coefficients given by Proposition 9, along with Mansour's calculations of the Fourier coefficients of the tribes function [Man98]. Note that the subsequent work [MO02] gives an elementary exact calculation of the noise stability of the tribes function, demonstrating that the calculation to follow is nearly tight.

Let us first rigorously define the tribes function. Note that the probability the tribes function is true is exactly $1 - (1 - 2^{-b})^{k/b}$. Since we want the tribes function to be as close to balanced as

possible, we need to define k and b in such a way that $(1-2^{-b})^{n/b} \approx 1/2$. In order to get this, we will actually define k in terms of b, so that T_k will only be defined on certain input lengths.

For every integer b, let $k = k_b$ be the smallest integral multiple of b such that $(1-2^{-b})^{k/b} \leq 1/2$. For analysis purposes, let k' be the real number such that $(1-2^{-b})^{k'/b} = 1/2$. Then $b = \log_2 k' - \log_2 \ln k' + o(1)$; hence $k' \leq k \leq k' + \log_2 k'$, and so $b = \log_2 k - \log_2 \ln k + o(1)$ as well. The probability that T_k is true on a random input is $(1-2^{-b})^{k/b} = (1/2)(1-2^{-b})^{(k-k')/b} = (1/2)(1-2^{-b})^{1+o(1)} = (1/2)(1 - O(\log k/k))$.

Proposition 14 *Let b and k be chosen as described, and let T_k be the associated tribes function. Then:*

$$\mathrm{NoiseStab}_\delta(T_k)^* \leq [\exp((1-\delta)^b) - 1] + O(\log^2 k/k^2).$$

Proof: We want to upper-bound $\mathrm{NoiseStab}_\delta(T_k)^*$ using Proposition 9, so we need to know the Fourier coefficients of T. We just showed that the probability T_k is true on a random input is $(1/2)(1 - O(\log k/k))$; hence $|\widehat{T_k}(\emptyset)| = O(\log k/k)$. Mansour studied the other Fourier coefficients of T_k in [Man98]; he showed that:

$$\begin{aligned} |\widehat{T_k}(S_1, \ldots, S_{k/b})| &= 2(1-2^{-b})^{k/b}(2^b-1)^{-j} \\ &\leq (2^b-1)^{-j}, \end{aligned}$$

where $S_i \subseteq \{x_{b(i-1)+1}, \ldots, x_{bi}\}$ and j is the number of i's such that $S_i \neq \emptyset$.

How many sets $S = (S_1, \ldots, S_{k/b})$ of cardinality d have exactly j nonempty parts S_i? There are $\binom{k/b}{j}$ choices for the nonempty parts, and at most $\binom{jb}{d}$ ways to distribute the elements of S among these parts. Hence there are at most $\binom{k/b}{j}\binom{jb}{d}$ such sets. Each set contributes $(1-2\delta)^d(2^b-1)^{-2j}$ to the sum $\mathrm{NoiseStab}_\delta(T_k)^* = \sum_{I \subseteq [k]} (1-2\delta)^{|S|}\widehat{T_k}(S)^2$. The case $k = 0$ we did separately; it contributes $O(\log^2 k/k^2)$ to the sum. Hence:

$$\begin{aligned} \mathrm{NoiseStab}_\delta(T_k)^* &\leq O(\log^2 k/k^2) + \sum_{j=1}^{k/b} \sum_{d=j}^{jb} \binom{k/b}{j}\binom{jb}{d}(1-2\delta)^d(2^b-1)^{-2j} \\ &\leq O(\log^2 k/k^2) + \sum_{j=1}^{k/b} \binom{k/b}{j}(2^b-1)^{-2j}(2-2\delta)^{jb} \\ &\leq O(\log^2 k/k^2) + \sum_{j=1}^{k/b} \frac{(k/b)^j}{j!}\frac{1}{[2^b(2^b-2)]^j}(2-2\delta)^{bj} \\ &= O(\log^2 k/k^2) + \sum_{j=1}^{k/b} \left[\frac{k}{b(2^b-2)}\right]^j (1-\delta)^{bj}/j! \end{aligned}$$

Since $(1-2^{-b})^{k/b} = 1/2 - o(1)$, $\frac{k}{b}\log_2(1-2^{-b}) = -1 - o(1)$. The Taylor approximation for $\log_2(1-x)$ lets us conclude that $\log_2(1-2^{-b}) \geq -2^{-b}/\ln 2$, so $\frac{k}{b2^b} \leq (\ln 2)(1+o(1)) < 1$. We conclude that $\frac{k}{b(2^b-2)} < 1$ asymptotically. Continuing our estimate, we get:

$$\begin{aligned} \mathrm{NoiseStab}_\delta(T_k)^* &\leq O(\log^2 k/k^2) + \sum_{j=1}^{k/b} (1-\delta)^{bj}/j! \\ &\leq O(\log^2 k/k^2) + [\exp((1-\delta)^b) - 1], \end{aligned}$$

as claimed. □

When δ is nearly $1/2$, the noise stability of T_k is almost as small as $1/2 + 1/k$:

Proposition 15 *For any $r > 0$, $\mathrm{NoiseStab}_{1/2-r}(T_k)^* \leq k^{-1+3r}$ for k sufficiently large.*

Proof: By Proposition 14,

$$\begin{aligned}
\mathrm{NoiseStab}_{1/2-r}(T_k)^* &\leq [\exp((1/2+r)^b) - 1] + O(\log^2 k/k^2) \\
&= [\exp((1/2)^b(1+2r)^b) - 1] + O(\log^2 k/k^2) \\
&\leq [\exp\big((1+o(1))\frac{\ln k}{k}(1+2r)^{\log_2 k}\big) - 1] + O(\log^2 k/k^2) \\
&= [\exp\big((1+o(1))\frac{\ln k}{k}\, k^{\log_2(1+2r)}\big) - 1] + O(\log^2 k/k^2) \\
&= [\exp(k^{-1+2r/\ln(2)+o(1)}) - 1] + O(\log^2 k/k^2) \\
&\leq 2k^{-1+2r/\ln(2)+o(1)} + O(\log^2 k/k^2) \\
&\leq k^{-1+3r},
\end{aligned}$$

for sufficiently large k, as $3 \geq 2/\ln(2)$. □

As an immediate corollary of Proposition 15 and Proposition 7, we get:

Proposition 16 *For every constant $\eta > 0$, there is a constant $r > 0$ such that $\mathrm{ExpBias}_{1-r}(T_k) \leq 1/2 + k^{-1/2+\eta}$ for k sufficiently large.*

We our now ready to prove our main theorem, a $1 - 1/\mathrm{poly}(n) \to 1/2 + n^{-1/2+\epsilon}$ hardness amplification for NP under the assumption that the initial hard function family in NP is balanced. In Section 8 we show how to get a result which is almost as strong without making the balance assumption.

Theorem 2 *If there is a family of functions (f_n) in NP which is infinitely often balanced and $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then there is a family of functions (h_m) in NP which is infinitely often $(\frac{1}{2} + n^{-1/2+\eta})$-hard for circuits of polynomial size, for any small $\eta > 0$.*

Proof: By Theorem 13, the hypothesis of the present theorem implies the existence of a family of functions (f_n) in NP which is infinitely often balanced and $(1/2+o(1))$-hard for polynomial circuits. We now proceed in a manner similar to that in the proof of Theorem 13.

Pick b such that the associated tribes input length $k = k_b$ is roughly n^C, where C is a large constant to be chosen later. Let $h_m = T_k \otimes f$, a function on $m = kn = n^{C+1}$ bits. Since T_k is monotone and in P, the family (h_m) is in NP. It remains to show that h_m is $(1/2 + m^{-1/2+\eta})$-hard for polynomial circuits whenever f_n is balanced and $(1/2 + o(1))$-hard.

Apply Theorem 1 with $\epsilon = 1/k$, δ set to the given $1/2 - o(1)$, and r a small positive constant to be chosen later. Then we conclude that h_m is $(\mathrm{ExpBias}_{(1-r)}(T_k) + \epsilon)$-hard for polynomial circuits. By Proposition 16, $\mathrm{ExpBias}_{(1-r)}(T_k)$ can be made smaller than $1/2 + k^{-1/2+\eta/4}$ by taking r sufficiently small. Since $\epsilon = 1/k$, the hardness of h_m is at most $1/2 + k^{-1/2+\eta/2}$ (for sufficiently large k). But $m = n^{C+1} = k^{1+1/C}$. Hence if we take C to be any constant greater than $1/\eta$ then $(k^{1+1/C})^{-1/2+\eta} \geq k^{(1+\eta)(-1/2+\eta)} = k^{-1/2+\eta/2+\eta^2} \geq k^{-1/2+\eta}$, and the proof is complete. □

This result more or less exhausts the possibilities for monotone hardness amplification via Theorem 1. The reason is that $\mathrm{NoiseStab}(g)^*$ will never be smaller than $\mathrm{polylog}(n)/n$ when g is monotone.

To be exact, a celebrated theorem of Kahn, Kalai, and Linial [KKL88] shows that for any monotone function g on k inputs, $\sum_{|S|=1} \widehat{g}(S)^2 \geq \Omega(\log^2 k/k)$. From Proposition 9, it immediately follows that $\mathrm{NoiseStab}_\delta(g)^* \geq (1-2\delta)\Omega(\log^2 k/k)$. If we start with a balanced function f on n inputs which has hardness at least $1/2+\Omega(\log^2 n/n)$ and then apply Theorem 1 with g as an amplifier, we end up with a function with input length nk and a hardness bound of $1/2+\Omega(\log^2 n/n)\Omega(\log^2 k/k) \geq 1/2 + \Omega(\log^2(nk)/nk)$.

Consequently, we cannot use these techniques to amplify hardness within NP below $1/2 + \Omega(\log^2 n/n)$. Further, from a constructive point of view, if we always use $\mathrm{NoiseStab}_\delta$ as an estimator for $\mathrm{ExpBias}_{2\delta}$ we pay the additional penalty of Proposition 7 and we will never do better than $1/2 + \tilde{\Omega}(n^{-1/2})$.

8 Eliminating the balance hypothesis

Theorem 2 requires that the initial hard function in NP is balanced on infinitely many of the input lengths for which it is hard. This assumption is not unrestrictive; the author is unaware of any NP-complete language which is balanced on all input lengths. We can remove the assumption at the expense of a slight loss in hardness in the final function:

Theorem 3 *If there is a family of functions (f_n) in NP which is infinitely often $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then there is a family of functions (h_m) in NP which is infinitely often $(\frac{1}{2} + m^{-1/3+\eta})$-hard for circuits of polynomial size, for any small $\eta > 0$.*

We devote the remainder of this section to proving Theorem 3. The proof only requires playing some tricks with input lengths. First, a simple trick using padding:

Proposition 17 *If there is a family of functions (f_n) in NP which is infinitely often $(1-1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then for every $\eta > 0$ there is a family of functions (g_m) in NP which is infinitely often $(1 - 1/m^\eta)$-hard for circuits of polynomial size.*

Proof: Suppose (f_n) is infinitely often $(1 - 1/n^c)$-hard for polynomial circuits. Pick $K > c/\eta$, and define $m = m(n) = n^K$. On an input x of length m, define $g_m(x)$ to be f applied to the first n bits of x. Then (g_m) is surely in NP, and we claim that whenever n is an input length for which f_n is $(1 - 1/n^c)$-hard for polynomial circuits, g_m is $(1 - 1/m^\eta)$-hard for polynomial size. The proof is simple; suppose C were a circuit of size m^d which correctly computed g_m on a $1 - 1/m^\eta$ fraction of the inputs in $\{0,1\}^m$. By an averaging argument, there are settings for the last $m - n$ input bits to C such that C correctly computes g_m on a $1 - 1/m^\eta$ fraction of the remaining inputs, $\{0,1\}^n$. But now we have a circuit of size $n^{Kd} = \mathrm{poly}(n)$ which computes f_n with probability at least $1 - 1/n^{\eta K} > 1 - 1/n^c$, a contradiction. □

Next, let's combine the tribes function and the recursive majority of 3 function to get a single function which is highly noise unstable under a small amount of noise:

Proposition 18 *For every $r > 0$ there is a monotone function family (F_k) in P defined for every input length k such that $\mathrm{NoiseStab}_{1/k^r}(F_k)^* \leq k^{-1+10r}$ for all sufficiently large k.*

Proof: Let $r' = 9r$. On input length k, pick ℓ as large as possible so that $3^\ell \leq k^{r'}$, and pick k' to be the largest possible tribes function input length which is at most $k^{1-r'}$. Define $F_k = T_{k'} \otimes \text{REC-MAJ-3}^\ell$ to be on k bits by ignoring some input bits if necessary. Note that $3^\ell = \Omega(k^{r'})$ and $k' = \Omega(k^{1-r'})$, so we haven't lost too much in the input length.

By Proposition 11, $\mathrm{NoiseStab}_{1/k^r}(\text{REC-MAJ-3}^\ell)^* \leq (k^r)^{1.1}/(3^\ell)^{.15} = O(k^{1.1r-.15r'}) = O(k^{-.25r}) \leq r/3$ for k sufficiently large. Now invoking Proposition 8 and Proposition 15, we conclude that $\mathrm{NoiseStab}_{1/k^r}(F_k)^* \leq (k^{1-r'})^{-1+3r/3} \leq k^{-1+10r}$, as claimed. □

We now outline the proof of Theorem 3. We begin with a $(1-1/n^\eta)$-hard function f on n inputs by using Proposition 17, and we wish to apply Theorem 1 using the function on k inputs defined in Proposition 18. The trouble is that the initial function has an unknown bias. To circumvent this, we map input instances of length n to various instance lengths around $L = L(n)$, and use the new input lengths as *guesses* for the bias of the initial hard function. This allows us to "know" the bias of f to within about $\pm 1/L$. At this point we can easily cook up a slightly altered version of f which is still hard and is within about $\pm 1/L$ of being balanced. This lets us apply Theorem 1, as long as $\epsilon/k \approx 1/L$. Since the expected bias we get from Proposition 18 is about $1/\sqrt{k}$, the largest ϵ we would like to pick is $1/\sqrt{k}$, leading to $1/k^{3/2} \approx 1/L \Rightarrow 1/\sqrt{k} \approx L^{-1/3}$. This is why the exponent of $1/3$ arises in Theorem 3. We now give the rigorous proof:

Proof: (of Theorem 3) Let $\eta > 0$ be given, and assume there is a family of functions (f_n) in NP that is infinitely often $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size. By Proposition 17, there is a family of functions in NP, which we will also denote by (f_n), that is infinitely often $(1 - 1/n^\eta)$-hard for polynomial circuits. We now begin to define h_m. Let $C = C(\eta)$ be a large constant divisible by 3 to be chosen later. On input x of length m, express $m = (n+1)^{C+1} + i$ for n as large as possible with $i \geq 0$. Assuming $0 \leq i < 8n^C$, we "guess" that the fraction of inputs on which f_n is 1 is about $i/8n^C$. (Note that i may be as large as $(n+2)^{C+1} - (n+1)^{C+1} > 8n^C$; when $i \geq 8n^C$, define h_m arbitrarily, say $h_m \equiv 0$.) Specifically, define $f'_n : \{0,1\}^{n+1} \to \{0,1\}$ as follows: on input yb, where $|y| = n$, $|b| = 1$, put

$$f'_n(yb) = \begin{cases} f(y) & \text{if } b = 0, \\ 0 & \text{if } b = 1 \text{ and } y \text{ is one of the lexicographically first } (i/8n^C)2^n \text{ strings in } \{0,1\}^n, \\ 1 & \text{else.} \end{cases}$$

The point of this construction is that for every n, there is a particular i and hence a particular m for which f'_n becomes very close to balanced; specifically, $\mathrm{bias}(f'_n) \leq 1/2 + 1/8n^C$. Further, note that f'_n is easily seen to be $(1 - 2/n^\eta)$-hard for polynomial circuits.

Now we continue the definition of h_m. Pick $k = n^{(2/3)C}$ and let F_k be the function from Proposition 18, with parameter $r > 3\eta/2C$ (with this choice, $2/n^\eta > 1/k^r$ for sufficiently large n). On input x of length m (where $m = (n+1)^{C+1} + i$ with $0 \leq i < 8n^C$), write $x = y_1 y_2 \cdots y_k z$, where $|y_i| = n+1$ and $|z| = m - k(n+1) > 0$. Now define $h_m(x) = F_k \otimes f'_n$, where the input bits z are ignored.

One easily checks that the family (h_m) is indeed in NP. We now show that (h_m) is infinitely often $(1/2 + m^{-1/3+O(1)\eta})$-hard for polynomial circuits, which is sufficient to complete the proof.

Suppose n is an input length for which f_n is hard. Then as noted there is an m for which f'_n becomes close to balanced, $\mathrm{bias}(f'_n) \leq 1/2 + 1/8n^C$. For this m, we claim that h_m has the desired hardness. This follows in a straightforward manner from Theorem 1, using Proposition 18 and Proposition 7. To be exact, let $\delta = 2/n^r$, $\epsilon = 1/n^{(1/3)C}$, and $r = 1$. Note that $\mathrm{bias}(f'_n) \leq 1/2 + 1/8n^C \leq (1-2\delta)\epsilon/4k$. Theorem 1 tells us that h_m is $(\mathrm{ExpBias}_\delta(F_k)+\epsilon)$-hard for polynomial circuits. By Propositions 7 and 18, $\mathrm{ExpBias}_\delta(F_k) \leq 1/2 + k^{-1/2+5(3\eta/2C)} + 1/n^{(1/3)C} = 1/2 + n^{-(1/3)C+5\eta} + n^{-(1/3)C}$. As a function of the input length, $m \leq (n+2)^{C+1}$, the quantity $n^{-(1/3)C+5\eta} + n^{-(1/3)C}$ can be made smaller than $m^{-1/3+O(1)\eta}$ by taking $C = O(1/\eta)$. □

Acknowledgements

Thanks to Madhu Sudan for suggesting both the problem and the approach, as well as for many helpful discussions. Russell Impagliazzo independently proved a $1 - 1/\text{poly} \to 1 - \Omega(1)$ hardness reduction for NP; many thanks to him for sharing his proof with me. Thanks to Adam Klivans for much explanation and encouragement; in particular, for emphasizing [Imp95] to me. Thanks also to Rocco Servedio, Elchanan Mossel, and Michael Rosenblum for helpful discussions and comments.

References

[Bez94] S. Bezrukov. Isoperimetric problems in discrete spaces. In *Extremal Problems for Finite Sets*, P. Frankl, Z. Furedi, G. Katona, D. Miklós eds. Bolyai Soc. Math. Stud. 3, 1994.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3, 1993.

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

[BKS99] I. Benjamini, G. Kalai, O. Schramm. Noise sensitivity of boolean functions and applications to percolation. *Inst. Hautes Études Sci. Publ. Math.*, 90, 1999.

[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.

[Fel68] W. Feller. An introduction to probability theory and its applications. John Wiley & Sons, 1968.

[GNW95] O. Goldreich, N. Nisan, A. Wigderson. On Yao's XOR-Lemma. *Electronic Colloquium on Computational Complexity*, 1995.

[GR00] M. Goldmann, A. Russell. Spectral bounds on general hard core predicates. *Symposium on Theoretical Aspects of Computer Science*, 2000.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. *Foundations of Computer Science*, 1995.

[IW97] R. Impagliazzo, A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. *Symposium on the Theory of Computation*, 1997.

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

[KS99] A. Klivans, R. Servedio. Boosting and hard-core sets. *Foundations of Computer Science*, 1999.

[LMN93] N. Linial, Y. Mansour, N. Nisan. Constant depth circuits, Fourier transforms, and learnability. *Journal of the ACM*, 40(3), 1993.

[Man98] Y. Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50(3), 1995.

[MO02] E. Mossel, R. O'Donnell. On the noise sensitivity of monotone functions. *Colloquium on Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities*, 2002.

[Saz81] V. Sazonov. Normal approximation — some recent advances. *Lecture Notes in Mathematics 879*, Springer-Verlag, 1981.

[Sha01] R. Shaltiel. Towards proving strong direct product theorems. *Electronic Colloquium on Computational Complexity*, 2001.

[STV01] M. Sudan, L. Trevisan, S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2), 2001.

[Yao82] A. Yao. Theory and application of trapdoor functions. *Foundations of Computer Science*, 1982.

A Proof of Lemma 5

Here we prove the strong version of Lemma 5. We slightly generalize by allowing p's range to be $[-1,1]$ rather than $[0,1]$. Also, because it makes notation much more compact, we call h's two-element range $\{+1,-1\}$ rather than $\{0,1\}$.

Lemma 5 *Let $\mathcal{B}_n$ denote the n-dimensional Hamming cube, and suppose $h : \mathcal{B}_n \to \{+1,-1\}$ and $p : \mathcal{B}_n \to [-1,1]$. Further suppose that $|\mathbf{E}[hp]| \geq |\mathbf{E}[h]| + \epsilon$. Then there exists an edge (x,y) in $\mathcal{B}_n$ such that $|p(x) - p(y)| \geq \Omega(\epsilon/\sqrt{n})$.*

Proof: Let δ be the maximum of $|p(x) - p(y)|$ over all edges (x,y) in $\mathcal{B}_n$. For a subset $C \subseteq \mathcal{B}_n$, let $\mu_p(C)$ denote the average value of p on C. Let m be the median value of p on $\mathcal{B}_n$, and partition $\mathcal{B}_n$ into two sets A^+ and A^- of equal size, such that $p(x) \geq m \geq p(y)$ for all $x \in A^+$ and $y \in A^-$.

We are interested in the the number of points in $\mathcal{B}_n$ at distance at most d from A^+. When $|A^+| = \frac{1}{2}2^n$, an isoperimetric inequality on the cube (see, e.g., [Bez94]) tells us that for *all* d this number is maximized when A^+ is a ball of radius $n/2$. (Strictly speaking, n should be odd and the radius should be $\lfloor n/2 \rfloor$. Since we only care about asymptotics in n, we gloss over such niceties.) It follows that the *average* distance from A^+ (over all points) is maximized when A^+ is such a ball.

The average distance in $\mathcal{B}_n$ to a ball of radius $n/2$ is $O(\sqrt{n})$, and this fact is fairly standard. To see it, suppose without loss of generality the ball is $A = \{x \in \mathcal{B}_n : \mathsf{MAJ}(x) = 0\}$. Then the distance from a point x to A is equal to 0 if $MAJ(x) = 0$, and is equal to $(\text{weight}(x) - n/2)$ otherwise. The result now follows from the fact that $\text{weight}(x)$ is distributed approximately as a normal variable $N(n/2, n/4)$, and that $\mathbf{E}[|N(n/2,n/4) - n/2|] = O(\sqrt{n})$.

Hence the average distance to A^+ over all points in $\mathcal{B}_n$ is $O(\sqrt{n})$. Indeed, since half of all points have distance 0 from A^+, we can conclude that the average distance to A^+, over any set of size $\frac{1}{2}2^n$, is $O(\sqrt{n})$ (gaining a factor of 2 in the $O(\cdot)$).

But if the Hamming distance between two points x and y is at most d, then $|p(x) - p(y)| \leq d\delta$. Since the smallest possible value of p on A^+ is m, it follows that $\mu_p(C) \geq m - O(\sqrt{n}\delta)$ for any set $|C| = \frac{1}{2}2^n$.

Running the same argument with A^- in place of A^+ yields that $\mu_p(C) \leq m + O(\sqrt{n}\delta)$ for any set $|C| = \frac{1}{2}2^n$. Writing $\theta = O(\sqrt{n}\delta)$, we conclude that $\mu_p(C) \in [m-\theta, m+\theta]$ for all sets $|C| = \frac{1}{2}2^n$.

Now let us turn our attention to h. Write $H^+ = h^{-1}(+1)$, $H^- = h^{-1}(-1)$ and assume without loss of generality that $b = 2^{-n}|H^+| \geq 1/2$. We first upper-bound $\mu_p(H^+)$. Let M be the set of $\frac{1}{2}2^n$ points in H^+ with largest p-value. Then $\mu_p(M) \leq m + \theta$. The remaining points in H^+ have p-value at most m. Hence:

$$\begin{aligned} \mu_p(H^+) &\leq \frac{1/2}{b}(m+\theta) + \frac{b-1/2}{b}m \\ \Rightarrow \quad b\mu_p(H^+) &\leq bm + \theta/2. \end{aligned}$$

Now we lower-bound $\mu_p(H^-)$. Let N be the set of $(b-\frac{1}{2})2^n$ points outside of H^- with smallest p-value. Then $|N \cup H^-| = \frac{1}{2}2^n$, so $\mu_p(N \cup H^-) \geq m - \theta$. On the other hand, the points in N have p-value at most m. Hence:

$$\begin{aligned} \frac{b-1/2}{1/2}m + \frac{1-b}{1/2}\mu_p(H^-) &\geq m - \theta \\ \Rightarrow (1-b)\mu_p(H^-) &\geq (1-b)m - \theta/2. \end{aligned}$$

Subtracting the two inequalities, we get:

$$\begin{aligned} b\mu_p(H^+) - (1-b)\mu_p(H^-) &\leq (2b-1)m + \theta \\ \Rightarrow \mathbf{E}[hp] &\leq \mathbf{E}[h]m + \theta \\ \Rightarrow \mathbf{E}[hp] &\leq |\mathbf{E}[h]| + \theta, \end{aligned}$$

since $m \leq 1$, and we assumed $\mathbf{E}[h] \geq 0$. Since we could replace p by $-p$ throughout, we can in fact conclude $|\mathbf{E}[hp]| \leq |\mathbf{E}[h]| + \theta$. But $|\mathbf{E}[hp]| \geq |\mathbf{E}[h]| + \epsilon$ by assumption. Hence $\theta \geq \epsilon$ which implies $\delta \geq \Omega(\epsilon/\sqrt{n})$. □

B A corresponding easiness theorem

In this section we demonstrate that Theorem 1 is close to being tight by showing how to construct, for any δ and any g, a balanced function f which is $(1-\delta)$-hard, yet such that $g \otimes f$ is roughly $\mathrm{ExpBias}_{2\delta}(g)$-easy (i.e., there *is* a circuit which computes the function with this probability). The constructions of the functions used in this section are essentially those of Shaltiel [Sha01].

Proposition 19 *There is a universal constant $\gamma \geq 1/8$ such that there exists a balanced function $h : \{0,1\}^n \to \{0,1\}$ which is $(1/2 + 2^{-\gamma n})$-hard for size $2^{\gamma n}$.*

Proof: Folklore proof, by picking h to be a random function. □

Definition 12 *For a given rational number δ, define a canonical constant z_δ and a canonical constant-sized circuit Z_δ on z_δ inputs such that $\Pr[Z_\delta = 0] = \delta$.*

Definition 13 *Given a function $f : \{0,1\}^n \to \{0,1\}$ and a rational constant δ, define $f_\delta : \{0,1\}^{n+z_\delta+1}$ by:*

$$f(x,a,b) = \begin{cases} f(x) & \text{if } Z_\delta(a) = 0 \\ b & \text{else} \end{cases}$$

Proposition 20

- *If f is balanced, so is f_δ.*
- *If f is η-hard for size s, then f_δ is $(1-\delta+\delta\eta)$-hard for size s.*
- *f_δ is $(1-\delta)$-easy for size $O(1)$.*

Definition 14 *If $g:\{0,1\}^n \to \{0,1\}$, define $\mathsf{SIZE}(g)$ to be the size of the smallest circuit computing g exactly, and define $\mathsf{R\text{-}B\text{-}SIZE}(g)$ (standing for "restriction-bias-size") to be the size of the smallest circuit which, on input a length-n restriction ρ, outputs the bit value towards which g_ρ is biased. (If g_ρ is balanced, the circuit is allowed to output either 0 or 1.)*

Of course, we can't expect $g \otimes f$ to be easy to compute if g itself is hard to compute. So intuitively, we think of $\mathsf{SIZE}(g)$ as being quite small. $\mathsf{R\text{-}B\text{-}SIZE}(g)$ may or may not be small. For many simple functions though, such as parity, threshold functions, or tribes functions, it is small. The easiness theorem we now prove, as a converse to Theorem 1, allows for a tradeoff, depending on the comparitive values of $\mathsf{SIZE}(g)$ and $\mathsf{R\text{-}B\text{-}SIZE}(g)$.

Theorem 21 *Let δ be a rational constant. Let $\delta' = \delta/(1-2^{-\gamma n}) = \delta + O(2^{-\gamma n})$, where γ is the constant from Proposition 19. Then there is a balanced function $f:\{0,1\}^{n+O(1)} \to \{0,1\}$ which is $(1-\delta)$-hard for circuits of size $2^{\gamma n}$, such that:*

1. *$g \otimes f$ is $\mathrm{ExpBias}_{2\delta'}(g)$-easy for size $\mathsf{R\text{-}B\text{-}SIZE}(g) + O(k)$*
2. *$g \otimes f$ is $\mathrm{NoiseStab}_{\delta'}(g)$-easy for size $\mathsf{SIZE}(g) + O(k)$*
3. *$g \otimes f$ is $(\mathrm{ExpBias}_{2\delta'}(g) - O(1/\sqrt{m}))$-easy for size $m\mathsf{SIZE}(g) + O(mk)$*

Proof: Take h to be the hard function from Proposition 19, and let $f = h_{2\delta'}$. Then by Proposition 20, f is balanced and $(1-\delta)$-hard for size $2^{\gamma n}$. But it is also $(1-2\delta')$-easy for size $O(1)$. We are now essentially in the Hard-Core scenario described in Section 2; with probability $(1-2\delta')$ we *know* the correct answer, and with probability $2\delta'$, we have no good idea.

Let's call our inputs $x_1, \ldots, x_k$, where $x_i = (x_i', a_i, b_i)$. Let A be the constant-sized circuit which on input $(x_i', a_i, b_i) \in \{0,1\}^{n+z_{\delta'}+1}$ outputs b_i if $Z_{\delta'}(a_i) \neq 0$, or $\star$ if $Z_{\delta'}(a_i) = 0$. Let B be the circuit of size $O(k)$ which applies a copy of A to each input x_i. The output of B is a length-k restriction, and we have the property that the output distribution of B is exactly that of $P^k_{2\delta'}$.

Now we apply the different guessing strategies suggested in Section 2, given the restriction ρ output by B.

To get result 1, we use a circuit of size $\mathsf{R\text{-}B\text{-}SIZE}(g)$ to output the more likely value of g_ρ over a uniform choice of inputs.

To get result 2, our circuit picks a random bit for each coordinate on which ρ is $\star$; then a $\mathsf{SIZE}(g)$ circuit outputs the appropriate value of g. An averaging argument lets us trade off the randomness for nonuniformity.

To get result 3, we try to guess the bit towards which g_ρ is biased by trying many random values. Specifically, we take m copies of the randomized circuit for result 2 (size $m\mathsf{SIZE}(g)$), and take their majority (size $O(mk)$). Again, the $O(mk)$ random bits can be traded for nonuniformity. Let $a = \mathrm{adv}(g_\rho)$, and assume without loss of generality that g_ρ is biased towards 0. The probability that the majority of m random outputs of g is 1 is at most $\eta = \exp(-\frac{a^2}{1+a}\frac{m}{4})$, using the Chernoff bound. Consequently, the probability the circuit is correct is at least $(1-\eta)(\frac{1}{2}+\frac{1}{2}a) + \eta(\frac{1}{2}-\frac{1}{2}a) = \frac{1}{2}+\frac{1}{2}a - a\eta = \mathrm{bias}(g_\rho) - a\eta$. It can be shown that $a\exp(-\frac{a^2}{1+a}\frac{m}{4}) \leq O(1/\sqrt{m})$ (simple calculus shows that the quantity on the left is maximized for $a = \Theta(1/\sqrt{m})$). Averaging over ρ, we get the claimed result. $\square$