Computational Applications Of Noise Sensitivity

by

Ryan William O'Donnell

B.Sc., Mathematics and Computer Science, University of Toronto (1999)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author . . . .
Department of Mathematics
May 2, 2003

Certified by . . . .
Madhu Sudan
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Rodolfo Ruben Rosales
Chair, Applied Mathematics Committee

Accepted by . . . .
Pavel Etingof
Chair, Department Committee on Graduate Students

Computational Applications Of Noise Sensitivity

by

Ryan William O'Donnell

Submitted to the Department of Mathematics
on May 2, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis is concerned with the study of the *noise sensitivity* of boolean functions and its applications in theoretical computer science. Noise sensitivity is defined as follows: Let f be a boolean function and let $\epsilon \in (0, \frac{1}{2})$ be a parameter. Suppose a uniformly random string x is picked, and y is formed by flipping each bit of x independently with probability ϵ. Then the noise sensitivity of f at ϵ is defined to be the probability that $f(x)$ and $f(y)$ differ.

In this thesis we investigate the noise sensitivity of various classes of boolean functions, including majorities and recursive majorities, boolean threshold functions, and monotone functions. Following this we give new complexity-theoretic and algorithmic applications of noise sensitivity:

- Regarding computational hardness amplification, we prove a general direct product theorem that tightly characterizes the hardness of a composite function $g \otimes f$ in terms of an assumed hardness of f and the noise sensitivity of g. The theorem lets us prove a new result about the hardness on average of NP: If NP is $(1 - \text{poly}(n))$-hard for circuits of polynomial size, then it is in fact $(\frac{1}{2} + o(1))$-hard for circuits of polynomial size.

- In the field of computational learning theory, we show that any class whose functions have low noise sensitivity is efficiently learnable. Using our noise sensitivity estimates for functions of boolean halfspaces we obtain new polynomial and quasipolynomial time algorithms for learning intersections, thresholds, and other functions of halfspaces. From noise sensitivity considerations we also give a polynomial time algorithm for learning polynomial-sized DNFs under the "Random Walk" model; we also give the first algorithm that learns the class of "junta" functions with efficiency better than that of the brute force algorithm.

- Finally, we introduce a new collective coin-flipping problem whose study is equivalent to the study of "higher moments" of the noise sensitivity problem. We prove several results about this extension, and find optimal or near-optimal choices for the coin-flipping function for all asymptotic limits of the parameters. Our techniques include a novel application of the reverse Bonami-Beckner inequality.

Thesis Supervisor: Madhu Sudan
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to express many thanks to my advisor, Madhu Sudan. Madhu provided me with endless assistance and with superb insights and suggestions throughout my time at MIT. He is everything one could ask for in an advisor.

I would like to thank my main collaborators and good friends, Adam Klivans, Elchanan Mossel, and Rocco Servedio. Fine fellows, all, and their technical ideas have informed nearly all aspects of my research and this thesis.

I would like to thank my girlfriend Zeynep Tolon for her encouragement and support over the years. Zeynep is a great person and a great source of inspiration.

Finally, I would like to thank my coauthors — Nader Bshouty, Adam Klivans, Elchanan Mossel, Oded Regev, Rocco Servedio, and Benny Sudakov — for their contributions to this thesis.

In addition...

In connection with the work on the noise sensitivity of monotone functions, Elchanan and I would like to thank Gil Kalai for encouraging us to write up the result, and Yuval Peres for interesting discussions. In connection with the work on hardness amplification, I would like to thank Madhu Sudan for the idea of using the majority function to amplify hardness within NP, Russell Impagliazzo for showing me his independent proof of a $1 - 1/\text{poly} \to 1 - \Omega(1)$ amplification for NP, and Michael Rosenblum for a helpful discussion. In connection with the work on the noise sensitivity of halfspaces, Adam, Rocco, and I would like to thank Yuval Peres for sharing his proof of the $O(\sqrt{\epsilon})$ upper bound with us. In connection with the work on learning juntas, Elchanan, Rocco, and I would like to thank Adam Kalai and Yishay Mansour for telling us about their prior results; I would also like to thank Oded Regev for letting me include his proof of Theorem 5.5.12 in this thesis. In connection with the work on cosmic coin flipping from [MO02a], Elchanan and I would like to thank Oded Schramm and Nati Srebro for many helpful ideas and suggestions.

This thesis is dedicated to my parents, Eric and Judy O'Donnell.

Bibliogrpahic note

Much of this research has been published already, and most of it was performed jointly with other researchers. The work on the noise sensitivity of monotone functions from Sections 3.6, 3.7, 3.8, and 3.9 is joint work with Elchanan Mossel [MO02b]. The work on the noise sensitivity of halfpsaces and on learning functions of halfspaces from Sections 3.5, 3.10, 3.11, 3.12, 5.3, and most of Section 5.2 is joint work with Adam Klivans and Rocco Servedio [KOS02]. The work on hardness amplification from Chapter 4, along with preliminary versions of Sections 3.4, 3.7, and 3.9, appeared in [O'D02]. The work on learning DNF from random walks from Section 5.4 is joint work with Nader Bshouty, Elchanan Mossel, and Rocco Servedio [BMOS03]. The work on learning juntas from Section 5.5 is joint work with Elchanan Mossel and Rocco Servedio [MOS03]. The work on the cosmic coin flipping problem from Chapter 6, except for the upper bound on the parties' success rate in Section 6.9, is joint work with Elchanan Mossel [MO02a]. The upper bound from Section 6.9 along with the isoperimetric inequality from Section 7.1 is part of ongoing work with Elchanan Mossel, Oded Regev, and Benny Sudakov [MORS03].

Contents

List of Figures

Chapter 1

Introduction

The subject of this thesis is the *noise sensitivity* of boolean functions. To motivate the concept, imagine an idealized election in which there are n voters and only two parties, called $+1$ and -1. For simplicity, suppose each voter independently casts a vote for $+1$ or -1 uniformly at random. An election scheme is a boolean function on n bits mapping the voters' votes to a winning party.

There are several properties we would want an election scheme $f\colon \{+1,-1\}^n \to \{+1,-1\}$ to have; for example, it should be balanced (equally often $+1$ and -1), and it should exhibit some degree of symmetry among the n voters. What will concern us in this example is the scheme's robustness to errors in the recording of the votes. Suppose that, independently for each voter, there is an ϵ chance that the vote is misrecorded ($+1$ is changed to -1 and vice versa). What is the probability that this affects the outcome of the election, as defined by f? That is, what is the probability that f applied to the recorded votes differs from f applied to the true votes?

This probability is precisely what we mean by the *noise sensitivity of f at ϵ*. Let us make a formal definition.

Definition 1.0.1 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a boolean function, let x be a uniformly random string from $\{+1,-1\}^n$, and let y be derived from x by flipping each bit independently with probability ϵ. Then the* noise sensitivity of f at ϵ *is defined to be:*

$$\mathsf{NS}_\epsilon(f) = \mathbf{P}[f(x) \neq f(y)].$$

In this thesis we will calculate and estimate the noise sensitivity of various boolean functions. We will also give applications of the study of noise sensitivity to problems in theoretical computer science.

1.1 A brief history of noise sensitivity in computer science

We begin by outlining the historical development of the noise sensitivity concept in theoretical computer science.

1.1.1 Kahn, Kalai, and Linial

The noise sensitivity of boolean functions was first studied in 1988 by Kahn, Kalai, and Linial [KKL88], although they did not use the phrase "noise sensitivity," nor did they study precisely the quantity in Definition 1.0.1. The most outstanding contribution of this paper was probably its demonstration of the power of using Fourier analysis and other analytic techniques in solving combinatorial problems about the boolean hypercube. In particular, Kahn et al. gave an ingenious application of the Bonami-Beckner inequality [Bon68, Bon70, Bec75] from classical harmonic analysis; this inequality proved extremely powerful in future analyses of noise sensitivity and Fourier coefficients [BKK$^+$92, Tal94, Raz95, FK96, BK97b, Fri98, FK98, BKS99, Bou99, Bou01, BJ02, KS02].

The main result proved in Kahn et al.'s work was a theorem stating that any balanced boolean function has a variable with "influence" $\Omega(\frac{\log n}{n})$. (For definitions of "influence" and other terms see Chapter 2.) They also considered problems of packings in the hypercube with extremal distance distribution properties, and, most relevantly for us, mixing times for random walks on the hypercube in which the initial distribution is uniform over a given set. This last problem can be stated thus: Given a boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$, consider $A = f^{-1}(-1)$, a subset of the hypercube. Suppose we take a standard random walk on the hypercube in which the initial distribution is uniform over A. What can be said about the speed with which the walk converges to the stationary (uniform) distribution in L^2 distance?

This question is very similar in nature to the question of noise sensitivity. One part of Kahn et al.'s problem amounts to determining the probability that a random walk of length ϵn, starting from A, ends up outside A; this is almost the same as asking the probability that a randomly chosen point in A, when it has each bit flipped independently with probability ϵ, ends up outside A. This latter quantity is precisely the noise sensitivity of f (up to a factor of $|f^{-1}(-1)|/2^n$). Kahn et al. already gave a formula for their noise sensitivity-like quantity in terms of Fourier coefficients (cf. Proposition 2.3.1), and showed that bounds on Fourier tails implied bounds on mixing time (cf. Proposition 2.3.2). Interestingly, Kahn et al. wrote that studying the mixing times of such random walks (and thus, implicitly, noise sensitivity) was the original motivation for their paper.

1.1.2 Håstad

The next major development in the study of noise sensitivity came independently in 1997 with Håstad's work [Hås97] on the NP-hardness of approximation problems (see also [Hås01]). This breakthrough paper gave several new optimal inapproximability results for problems such as MAX-3SAT, MAX-3LIN, and Set Splitting. These results were applications of a new technique Håstad developed for designing and analyzing probabilistically checkable proofs (PCPs); the technique, which we now briefly describe, involves noise sensitivity in an essential way.

A crucial step in designing efficient PCPs is to encode certain information in an error correcting code called the "long code" (introduced in [BGS98]). The long code over a set of n objects has as its message space the set of all functions $f\colon \{+1,-1\}^n \to \{+1,-1\}$; the encoding of a particular object $i \in [n]$ is the "dictator" (projection) function $x \mapsto x_i$. Typically one has oracle access to a function f which is supposed to be a dictator function, and one wants to verify the fact that f depends on only one variable by making an extremely small number of queries to the oracle. Håstad showed that it suffices for applications to come up with a test which only checks that there is some small set of variables which significantly control the value of f. Håstad then gave such a test; although he did not phrase it in these terms, essentially his test works by performing one sample of the noise sensitivity of the unknown function. Roughly speaking, one chooses x and y as in Definition 1.0.1, queries f at these two points, and accepts if $f(x) = f(y)$. As did Kahn, Kalai, and Linial, Håstad analyzed his test by deriving and working with a formula for noise sensitivity in terms of Fourier coefficients (cf. Proposition 2.3.1).

In addition to deriving extremely important results via noise sensitivity, Håstad's work pointed the way to some of the more exciting current developments in noise sensitivity.

These include more PCP-based results, as well as tighter relationships between having low noise sensitivity and having the property of depending on only a small number of variables. We will discuss the latter more in Section 1.2.

1.1.3 Benjamini, Kalai, and Schramm

In 1998, Benjamini, Kalai, and Schramm [BKS99] gave the first comprehensive study of the noise sensitivity of boolean functions. In addition to introducing the quantity (which they too did not explicitly call "noise sensitivity") and the basic formulas, they gave several interesting new definitions and theorems. They defined a balanced family of functions $\{f_n\}$ to be "asymptotically noise sensitive" if $\mathsf{NS}_{.1}(f_n) \to \frac{1}{2}$ as $n \to \infty$ (this turns out not to depend on the value .1), and they defined it to be "asymptotically noise stable" if $\mathsf{NS}_\epsilon(f_n)$ can be upper-bounded by a function of ϵ alone which goes to 0 as $\epsilon \to 0$.

Benjamini et al. gave equivalent conditions that function families be asymptotically noise sensitive or noise stable in terms of Fourier coefficients. They also gave a deep theorem showing that if $I\!I(f_n) \to 0$ then $\{f_n\}$ is asymptotically noise sensitive (see Definition 2.3.6 for the definition of $I\!I(f)$). Using this theorem, Benjamini et al. showed that a certain family of functions defined by a bond percolation scenario is asymptotically noise sensitive. The authors also related asymptotic noise sensitivity and stability to correlations with weighted majority functions (i.e., boolean halfspaces), and showed that the class of weighted majority functions is asymptotically noise stable (cf. the results in Section 3.10 of this thesis). Finally, the authors also made some speculative conjectures connecting the noise stability of a function to the size and depth of the AC^0 and TC^0 circuits needed to compute it.

1.1.4 Other relevant works

We now briefly describe some other papers of relevance. Ben-Or and Linial [BL90] introduced the problem of collective coin-flipping in the perfect information model; we study a related problem in Chapter 6. They also studied the influence of variables on boolean functions, influencing and motivating the work of Kahn, Kalai, and Linial. Bourgain, Kahn, Kalai, Katznelson, and Linial [BKK+92] generalized some of the results in [KKL88], extending their theorem on the existence of a variable with influence $\Omega(\frac{\log n}{n})$ to functions on general product spaces, $f\colon X^n \to \{+1, -1\}$ (see also [Fri02]). Bshouty, Jackson, and Tamon [BJT99b] related noise sensitivity to computational learning theory under noise.

Schramm and Tsirelson [ST99] studied noise sensitivity on trees, as opposed to the boolean hypercube. Some works of Talagrand (e.g., [Tal96]) and Tsirelson (e.g., [Tsi99]) also have some relevance.

1.2 Current complexity-theoretic motivations — juntas

Håstad's work demonstrated the importance of a certain relaxation in long code-based probabilistically checkable proofs: instead of having to test whether a codeword function is close to a dictator function, it is often enough to test whether it is close to a function dependent on only a small number of coordinates. A function over many variables which depends on only a small number of them is known as a *junta*:

Definition 1.2.1 *A boolean function* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is called a* k-junta *if it only depends on some* k *of its* n *inputs.* f *is called a* (γ, k)-junta*, or is said to be* γ-close to a k-junta*, if there is a* k*-junta* $f'\colon \{+1,-1\}^n \to \{+1,-1\}$ *such that* f *and* f' *differ on at most* $\gamma 2^n$ *inputs.*

The terms "junta" and "dictator function" come from the theory of social choice (see, e.g., [Kal02]), and can be understood in terms of the election example described at the beginning of this chapter — a junta is a choice function that is controlled by a small number of voters. The same considerations explain why a 1-junta is called a dictator function.

Coming up with simple tests that determine whether a boolean function is close to being a junta has proved to be a very important problem for hardness of approximation results. Håstad's analysis showed that a function with low noise sensitivity must have some small set of coordinates which exhibit a reasonably significant amount of influence over the function. Håstad asked whether in fact having low noise sensitivity implied being close to a junta. In 1998 Friedgut [Fri98] showed that having low *average sensitivity* (see Definition 2.3.4 and Theorem 2.4.6) was a sufficient condition for being close to a junta. This result played an important role in the breakthrough paper of Dinur and Safra [DS02] which showed that the vertex cover problem is NP-hard to approximate to within a factor of 1.36. Friedgut's characterization was not necessary, however, as functions which are close to juntas may have very large average sensitivity. Finally, in 2001 Bourgain [Bou01] showed that indeed any function with sufficiently low noise sensitivity must be close to a junta. Bourgain's main theorem can be rephrased in terms of a corollary:

Theorem 1.2.2 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a boolean function, and let $\delta = \mathsf{NS}_\epsilon(f)$. If $\delta \leq \epsilon^{\frac{1}{2}+\Omega(1)}$, then f is a $(O(\delta), O(\delta^{-3})16^{\delta^{-4}})$-junta.*

The characterization is also necessary in a limited sense of the parameters; it is easy to show (using, say, Proposition 3.1.2) that any $(O(\delta), O(\delta^{-3})16^{\delta^{-4}})$-junta must have noise sensitivity at most $O(\delta)$ at $\epsilon = o(\delta^4 16^{-\delta^{-4}})$.

Bourgain's theorem may prove very useful for future PCP-based hardness of approximation results. A paper of Khot [Kho02] shows how Bourgain's theorem along with Khot's "Unique Games" conjecture can be used to show that it is NP-hard to distinguish between instances of 2-Linear Equations Mod 2 in which either a $1-\epsilon$ fraction of the equations is satisfiable or at most a $1-\epsilon^{\frac{1}{2}+\delta}$ fraction is satisfiable. (Khot attributes the result as being "essentially" due to Håstad.) We also note that Kindler and Safra [KS02] extended Bourgain's theorem (with weaker parameters) to the case of biased product measures on $\{+1,-1\}^n$; such generalizations were essential for the hardness result of Dinur and Safra.

Finally, we note that finding tests for juntas in the domains of property testing algorithms [PRS01, FKR$^+$02, CG02] and learning theory [GTT99] (see also Section 5.5) is an active research area.

1.3 New applications, and the outline of this thesis

In this thesis we give several new complexity-theoretic and algorithmic applications of noise sensitivity.

We begin in Chapter 2 by giving some basic facts about noise sensitivity, introducing the families of boolean functions we consider in the thesis, discussing the relationship between noise sensitivity and Fourier analysis, and introducing the Bonami-Beckner inequality.

In Chapter 3 we analyze and estimate the noise sensitivity of several families of boolean functions, including majority functions, recursive majorities, boolean threshold functions, tribes functions, and read-once intersections and majorities of boolean halfspaces. We also give an essentially complete picture of the maximal noise sensitivity of monotone functions for all ranges of ϵ.

In Chapter 4 we study the problem of *hardness amplification.* Hardness amplification asks the following question: Suppose a boolean function f on n inputs cannot be correctly computed on more than $(1-\epsilon)2^n$ inputs by circuits of a given size. Let g be another boolean function on m inputs, and consider the function h given by applying f to m independent strings of length n, and applying g to the resulting m outputs. How hard is h to compute, in terms of ϵ and properties of g, for circuits of comparable size? We give a near-tight answer to this question in terms of the noise sensitivity of g at ϵ. Using our answer and the analysis of the noise sensitivity of monotone functions, we show the following theorem regarding the hardness of NP: If NP is $(1 - \text{poly}(n))$-hard for circuits of polynomial size, then it is in fact $(\frac{1}{2} + n^{-\frac{1}{2}+o(1)})$-hard for circuits of polynomial size.

In Chapter 5 we turn our attention to computational learning theory. We first show how known techniques can be combined to prove that functions with low noise sensitivity can be efficiently learned under the uniform distribution. Using our noise sensitivity estimates for functions of boolean halfspaces we obtain the first known polynomial and quasipolynomial time algorithms for learning intersections, thresholds, and other functions of halfspaces. We next show that the notorious class of functions with polynomial-sized DNF formulas can be learned efficiently under the "Random Walk" model; the proof uses estimations of noise sensitivity-type quantities. Finally, we attack the same problem under the usual uniform distribution model by considering the class of junta functions, which as described in Section 1.2 are intimately involved in the study of noise sensitivity. We make the first known substantial progress over the trivial algorithm for learning juntas.

In Chapter 6 we introduce a new problem called the "cosmic coin-flipping problem," whose study is equivalent to the study of "higher moments" of the noise sensitivity problem. In the cosmic coin-flipping problem, a number of non-communicating parties wish to agree on a uniformly random bit. The parties have access to the same uniformly random "cosmic" string, but each sees only an ϵ-noisy version of it. Each party applies a boolean function to the bits it sees; we wish to find the functions that maximize the probability that all parties agree on their outputs. We prove several results about this extension of the noise sensitivity problem, and find optimal and near-optimal choices for the functions for all asymptotic limits of the parameters. One of our results uses the *reverse* Bonami-Beckner inequality, a technique which does not appear to have been previously exploited in the study of boolean functions.

Finally, in Chapter 7 we present some conjectures and open problems suggested by the work in this thesis.

Chapter 2

Definitions And Preliminaries

2.1 Noise sensitivity definitions

Throughout this thesis we shall write bits as $+1$ and -1 rather than the usual 0 and 1; this simplifies notation greatly, mainly because it means the parity of some bits is equal to their product. We view $+1$ as being "false" and -1 as being "true," and we sometimes write F and T in place of these quantities for clarity.

We shall also view the boolean hypercube $\{+1, -1\}^n$ as a probability space under the uniform probability measure $\mathbf{P}$. Unless otherwise indicated, $\mathbf{E}$ shall denote expectation over the uniform distribution.

Consider two different ways of making a bit $x_0 \in \{+1, -1\}$ noisy:

Definition 2.1.1 *We define* flipping *a bit $x_0 \in \{+1, -1\}$ as replacing x_0 with $-x_0$. We define* updating *x_0 as replacing x_0 with a uniform random bit.*

Note that the probabilistic experiments of updating a bit and flipping it with probability $\frac{1}{2}$ are identical. Let us introduce some notation for flipping a bit of a string:

Definition 2.1.2 *Given $x \in \{+1, -1\}$ and $j \in [n]$, we define $\sigma_j x$ to be x with its jth bit flipped.*

We now describe a noise operator which acts on strings:

Definition 2.1.3 *Given $x \in \{+1, -1\}^n$ and $0 \leq \epsilon \leq 1$, we define $N_\epsilon(x)$ to be the random variable taking values in $\{+1, -1\}^n$ given by flipping each bit of x independently with prob-*

ability ϵ. For $\epsilon \leq \frac{1}{2}$, $N_\epsilon(x)$ is equivalently defined by updating each bit in x independently with probability 2ϵ.

As special cases, note that $N_0(x)$ is the constant random variable x, $N_{\frac{1}{2}}(x)$ is a uniform random string independent of x, and $N_1(x)$ is constantly the string with Hamming distance n from x. We will almost always consider only $\epsilon \leq \frac{1}{2}$.

In the study of noise sensitivity, we consider choosing a uniformly random string x and letting y being an ϵ-noisy version of x. We will sometimes use the following facts:

Fact 2.1.4 *Let $x \in \{+1, -1\}^n$ be chosen uniformly at random and let $y = N_\epsilon(x)$. Then*

- *(x, y) has the same probability distribution as (y, x); and,*
- *the random variables $x_i y_i$ are mutually independent for $i \in [n]$, and $\mathbf{E}[x_i y_i] = 1 - 2\epsilon$.*

We now repeat Definition 1.0.1 using our new notation:

Definition 2.1.5 *Given a boolean function $f\colon \{+1, -1\}^n \to \{+1, -1\}$ and $0 \leq \epsilon \leq 1$, the* noise sensitivity of f at ϵ *is*

$$\mathsf{NS}_\epsilon(f) = \mathop{\mathbf{P}}_{x,\ y=N_\epsilon(x)}[f(x) \neq f(y)].$$

Since $f(x) \neq f(y) \Rightarrow f(x)f(y) = -1$, and $f(x) = f(y) \Rightarrow f(x)f(y) = +1$, we have the following:

Fact 2.1.6

$$\mathsf{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2} \mathop{\mathbf{E}}_{x,\ y=N_\epsilon(x)}[f(x)f(y)].$$

Since (x, y) and (y, x) have the same probability distribution, we have, for example, $\mathsf{NS}_\epsilon(f) = 2\mathbf{P}[f(x) = \mathrm{T}, f(N_\epsilon(x)) = \mathrm{F}]$. Consequently,

Fact 2.1.7

$$\begin{aligned} \mathsf{NS}_\epsilon(f) &= 2\,\mathbf{P}[f(x) = \mathrm{T}]\,\mathbf{P}[f(N_\epsilon(x)) = \mathrm{F} \mid f(x) = \mathrm{T}] \\ &= 2\,\mathbf{P}[f(x) = \mathrm{F}]\,\mathbf{P}[f(N_\epsilon(x)) = \mathrm{T} \mid f(x) = \mathrm{F}]. \end{aligned}$$

The noise sensitivity of any function at 0 is 0. Since x and $N_{\frac{1}{2}}(x)$ are independent, the noise sensitivity of f at $\frac{1}{2}$ depends only on the fraction of points on which f is -1; by Fact 2.1.6, $\mathsf{NS}_{\frac{1}{2}}(x) = \frac{1}{2} - \frac{1}{2}\mathbf{E}[f(x)]^2$. This quantity arises frequently:

Fact 2.1.8 *If $f\colon \{+1,-1\}^n \to \{+1,-1\}$ is a boolean function,*

$$\mathrm{Var}[f] = 1 - \mathbf{E}[f(x)]^2 = 2\,\mathsf{NS}_{\frac{1}{2}}(x) = 4\,\mathbf{P}[f = \mathrm{T}]\,\mathbf{P}[f = \mathrm{F}].$$

For fixed f, we have some general facts about $\mathsf{NS}_\epsilon(f)$ as a function of ϵ on $[0, \frac{1}{2}]$:

Proposition 2.1.9 *For any $f\colon \{+1,-1\}^n \to \{+1,-1\}$, $\mathsf{NS}_\epsilon(f)$ is a continuous, increasing, log-concave function of ϵ on $[0, \frac{1}{2}]$ which is 0 at $\epsilon = 0$ and $\frac{1}{2}\mathrm{Var}[f]$ at $\epsilon = \frac{1}{2}$. If f is nonconstant then $\mathsf{NS}_\epsilon(f)$ is strictly increasing in ϵ.*

The fact that $N_\epsilon(f)$ is continuous, increasing, and log-concave will follow immediately from Proposition 2.3.1.

Finally, we remark that adding irrelevant input variables to a function does not change its noise sensitivity.

2.2 Functions we consider

In this section we describe the sorts of functions whose noise sensitivity we will analyze. We begin with some very basic functions:

Definition 2.2.1

- $\mathsf{PARITY}_n\colon \{+1,-1\}^n \to \{+1,-1\}$ *is the parity function,* $(x_1, \dots, x_n) \mapsto \prod_{i=1}^n x_i$.
- $\pi_n^i\colon \{+1,-1\}^n \to \{+1,-1\}$ *is the function* $(x_1, \dots, x_n) \mapsto x_i$. *In the context of the study of boolean functions it is traditional to call the functions* $\{\pm\pi_n^i\colon i \in [n]\}$ *the* dictator functions *rather than the more usual "projection functions."*
- $\mathsf{AND}_n\colon \{+1,-1\}^n \to \{+1,-1\}$ *is the logical AND function, which is* T *(*-1*) if and only if all input bits are* T *(*-1*).*
- $\mathsf{OR}_n\colon \{+1,-1\}^n \to \{+1,-1\}$ *is the logical OR function, which is* T *(*-1*) unless all input bits are* F *(*$+1$*).*

- *For n odd,* $\mathsf{MAJ}_n\colon \{+1,-1\}^n \to \{+1,-1\}$ *is the majority function, definable as* $\mathsf{MAJ}_n(x_1,\ldots,x_n) = \mathrm{sgn}(\sum_{i=1}^n x_i)$. *We emphasize that this function is defined only for n odd.*

We sometimes drop the subscript n from these function names for clarity.

Aside from PARITY, all of the functions defined above are examples of *threshold functions*:

Definition 2.2.2 *A boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ is said to be a* (weighted) threshold function *or a* (boolean) halfspace *if there are numbers $w_1,\ldots,w_n \in \mathbf{R}$, $\theta \in \mathbf{R}$ such that*

$$f(x_1,\ldots,x_n) = \mathrm{sgn}\left(\left(\sum_{i=1}^n w_i x_i\right) - \theta\right).$$

The numbers w_i are called the weights *and the number θ is called the* threshold.

We will take $\mathrm{sgn}(0)$ to be undefined; consequently we will only admit threshold functions with the property that $\sum w_i x_i \neq \theta$ for every choice of $x \in \{+1,-1\}^n$. It is easily seen that this is without loss of generality, since the weights can be perturbed slightly to prevent equality.

We now describe some properties that boolean functions may possess.

Definition 2.2.3 *A boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ is said to be* balanced *if it is equally often $+1$ or -1 (i.e., if $\mathbf{E}[f] = 0$). It is said to be* antisymmetric *if $f(-x_1,\ldots,-x_n) = -f(x_1,\ldots,x_n)$ for all $x \in \{+1,-1\}^n$.*

Note that antisymmetric functions are necessarily balanced, and that any threshold function with threshold 0 is antisymmetric.

Definition 2.2.4 *A boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ is said to be* monotone *if $f(x) \le f(y)$ whenever $x \le y$ in the sense that $x_i \le y_i$ for all $i \in [n]$. Equivalently, f is monotone if changing input bits from $+1$ to -1 can only change the value of f from $+1$ to -1.*

Note that any threshold function in which all weights are nonnegative is monotone. This includes, e.g., π^i for all i, AND, OR, and MAJ. Functions which become monotone when the sense of $+1$ and -1 is switched for some input coordinates are called *unate*:

Definition 2.2.5 *A boolean function* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is said to be* unate *if there are signs* $s_1,\dots,s_n \in \{+1,-1\}$ *such that the function* $(x_1,\dots,x_n) \mapsto f(s_1x_1,\dots,s_nx_n)$ *is monotone.*

All threshold functions are unate.

Finally, we consider a simple operator which allows us to combine functions:

Definition 2.2.6 *If* $f\colon \{+1,-1\}^m \to \{+1,-1\}$ *and* $g\colon \{+1,-1\}^n \to \{+1,-1\}$ *are boolean functions, we define* $f \otimes g\colon \{+1,-1\}^{mn} \to \{+1,-1\}$ *by*

$$f \otimes g(x_1,\dots,x_{mn}) = f(g(x_1,\dots,x_n), g(x_{n+1},\dots,x_{2n}),\dots,g(x_{(m-1)n+1},\dots,x_{mn})).$$

We also recursively define $g^{\otimes^k}\colon \{+1,-1\}^{n^k} \to \{+1,-1\}$ *by* $g^{\otimes^1} = g$, $g^{\otimes^k} = g \otimes (g^{\otimes^{k-1}})$.

Note that the classes of monotone functions and unate functions are both closed under $\otimes$. We call a function of the form $\mathsf{OR}_m \otimes \mathsf{AND}_n$ a *read-once DNF* and a function of the form $\mathsf{AND}_m \otimes \mathsf{OR}_n$ a *read-once CNF*. We call a function of the form $\mathsf{MAJ}_n^{\otimes^k}$ a *recursive, or iterated, majority of n (of depth k)*.

We end this section with two simple but useful facts; the first follows immediately from definitions, the second from the union bound.

Proposition 2.2.7 *Suppose* $f\colon \{+1,-1\}^m \to \{+1,-1\}$, $g\colon \{+1,-1\}^n \to \{+1,-1\}$, *and* g *is a balanced function. Then* $\mathsf{NS}_\epsilon(f \otimes g) = \mathsf{NS}_{\mathsf{NS}_\epsilon(g)}(f)$.

Proposition 2.2.8 *Suppose* $f\colon \{+1,-1\}^m \to \{+1,-1\}$ *and* $g_i\colon \{+1,-1\}^{n_i} \to \{+1,-1\}$ *for* $i = 1\dots m$. *Then* $\mathsf{NS}_\epsilon(f(g_1,\dots,g_m)) \le \sum_{i=1}^m \mathsf{NS}_\epsilon(g_i)$.

2.3 Fourier analysis on $\{+1,-1\}^n$

It is often very helpful to relate the noise sensitivity of boolean functions to their Fourier coefficients. In this section we recall briefly the necessary notions from harmonic analysis on the hypercube. For a fuller treatment, see, e.g., [Šte02].

The probability space $\{+1,-1\}^n$ naturally gives rise to an inner product space on all functions $f\colon \{+1,-1\}^n \to \mathbf{R}$, with

$$\langle f, g\rangle = \mathbf{E}[fg] = 2^{-n} \sum_{x\in\{+1,-1\}^n} f(x)g(x).$$

For a set $S \subseteq [n]$ we define $\chi_S\colon \{+1,-1\}^n \to \{+1,-1\}$ to be the parity function $\chi_S(x) = \prod_{i\in S} x_i$. We also write simply x_S for $\chi_S(x)$. Note that $\mathbf{E}[\chi_S] = 0$ for $S \neq \emptyset$; further, $\chi_S\chi_{S'} = \chi_{S\Delta S'}$, where Δ denotes symmetric difference. It follows that $(\chi_S)_{S\subseteq[n]}$ is an orthonormal basis for the space of functions $f\colon \{+1,-1\}^n \to \mathbf{R}$. We define the *$S$ Fourier coefficient of f* to be the correlation of f with the parity function on S, $\hat{f}(S) = \langle \chi_S, f\rangle$. We have the *Fourier expansion of f*, $f(x) = \sum_{S\subseteq[n]} \hat{f}(S)x_S$ By Parseval's identity, for any boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ we have $\sum_{S\subseteq[n]} \hat{f}(S)^2 = 1$.

Because the basis $(\chi_S)_{S\subseteq[n]}$ has very nice properties with respect to the noise operator, we can get a formula for noise sensitivity in terms of Fourier coefficients:

Proposition 2.3.1

$$\mathsf{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2}\sum_{S\subseteq[n]} (1-2\epsilon)^{|S|}\hat{f}(S)^2.$$

Proof: Let x be chosen uniformly at random, and let $y = N_\epsilon(x)$. Now write f's Fourier expansion in Definition 2.1.6:

$$\begin{aligned}
\mathsf{NS}_\epsilon(f) &= \frac{1}{2} - \frac{1}{2}\mathbf{E}[f(x)f(y)] \\
&= \frac{1}{2} - \frac{1}{2}\mathbf{E}\left[\left(\sum_{S\subseteq[n]} \hat{f}(S)x_S\right)\left(\sum_{T\subseteq[n]} \hat{f}(T)y_T\right)\right] \\
&= \frac{1}{2} - \frac{1}{2}\sum_{S,T\subseteq[n]} \hat{f}(S)\hat{f}(T)\mathbf{E}[x_S y_T].
\end{aligned}$$

If $S \neq T$ then $\mathbf{E}[x_S y_T] = 0$. To see this, suppose, say, $i \in S \setminus T$; then $\mathbf{E}[x_S y_T] = \mathbf{E}[x_i]\mathbf{E}[x_{S\setminus\{i\}} y_T]$ by independence (see Fact 2.1.4) and $\mathbf{E}[x_i] = 0$. On the other hand, if $S = T$ then $\mathbf{E}[x_S y_T] = \mathbf{E}[x_S y_S] = \prod_{i\in S}\mathbf{E}[x_i y_i]$ again by independence, and the result follows as $\mathbf{E}[x_i y_i] = 1 - 2\epsilon$ by Fact 2.1.4. □

It is now immediate that $\mathsf{NS}_\epsilon(f)$ is a continuous, increasing, log-concave function of ϵ on $[0, \frac{1}{2}]$ since each summand $\hat{f}(S)^2(1-2\epsilon)^{|S|}$ is continuous, decreasing, and log-convex.

Further, if f is not constant then there is at least one $S \neq \emptyset$ such that $\hat{f}(S) > 0$ and hence $\mathsf{NS}_\epsilon(f)$ is strictly increasing for $\epsilon \in [0, \frac{1}{2}]$, as claimed in Proposition 2.1.9.

The formula in Proposition 2.3.1 provides an essential tool for studying and applying noise sensitivity; as noted in Chapter 1, it was discovered — in slightly different versions each time — by each of Kahn et al., Håstad, and Benjamini et al. The formula shows that the sensitivity of f to noise is closely related to how much of the ℓ_2 weight of f's Fourier coefficients is concentrated on sets S of large size. Since the weights $\hat{f}(S)^2$ are nonnegative and must sum to 1, we see that the more mass on coefficients with large $|S|$, the more noise sensitive f is. Indeed, all the authors mentioned noticed that upper bounds on noise sensitivity of f imply upper bounds on the tail of f's Fourier spectrum; thus the following result can be considered folklore:

Proposition 2.3.2 *For any $f\colon \{+1,-1\}^n \to \{+1,-1\}$, $\epsilon \in (0, \frac{1}{2}]$,*

$$\sum_{|S| \geq 1/\epsilon} \hat{f}(S)^2 < \frac{2}{1-e^{-2}}\, \mathsf{NS}_\epsilon(f).$$

Proof: Starting from Proposition 2.3.1, we have

$$\begin{aligned}
2\mathsf{NS}_\epsilon(f) &= 1 - \sum_{S \subseteq [n]} (1-2\epsilon)^{|S|} \hat{f}(S)^2 \\
&= \sum_{S \subseteq [n]} \hat{f}(S)^2 - \sum_{S \subseteq [n]} (1-2\epsilon)^{|S|} \hat{f}(S)^2 \\
&= \sum_{S \subseteq [n]} [1 - (1-2\epsilon)^{|S|}] \hat{f}(S)^2 \\
&\geq \sum_{|S| \geq 1/\epsilon} [1 - (1-2\epsilon)^{|S|}] \hat{f}(S)^2 \\
&\geq \sum_{|S| \geq 1/\epsilon} [1 - (1-2\epsilon)^{1/\epsilon}] \hat{f}(S)^2 \\
&> \sum_{|S| \geq 1/\epsilon} (1 - e^{-2}) \hat{f}(S)^2,
\end{aligned}$$

and the proof is complete. □

By considering parity functions, one sees that $\frac{2}{1-e^{-2}}$ is the sharpest possible constant that does not depend on n or ϵ. We will sometimes use the above result in the following alternate form which tells us just "how far up we need to go" to get all but δ of the Fourier spectrum:

Corollary 2.3.3 *Let* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *be any boolean function, and suppose* $m\colon [0,\frac{1}{2}] \to [0,1]$ *be a strictly increasing continuous function such that* $\mathsf{NS}_\epsilon(f) \le m(\epsilon)$*. Then*

$$\sum_{|S|\ge k} \hat{f}(S)^2 \le \delta \qquad \textit{for} \qquad k = \frac{1}{m^{-1}(\delta/2.32)}.$$

Proof: Immediate from Proposition 2.3.2, using $\frac{2}{1-e^{-2}} < 2.32$. □

We now consider a few more quantities derivable from a function's Fourier coefficients. Ben-Or and Linial [BL90] introduced notation for the influence of a variable on a boolean function (the quantity is known as the Banzhaf power index [Ban65] in the theory of choice); Kahn et al. [KKL88] introduced the notion of average sensitivity.

Definition 2.3.4 *Given* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *and* $j \in [n]$*, define*

$$I_j(f) = \mathop{\mathbf{P}}_{x\in\{+1,-1\}^n}[f(x) \ne f(\sigma_j x)],$$

called the influence of j on f*. Also define*

$$I(f) = \sum_{j=1}^{n} I_j(f),$$

called the average sensitivity of f*.*

Note that if f is monotone then $I_j(f)$ is precisely $\hat{f}(\{j\})$. The name "average sensitivity" comes from the fact that $I(f)$ is the expected number of indices in a uniformly randomly chosen x whose flipping causes the value of f to flip. Kahn et al. related $I(f)$ to the Fourier coefficients of f (monotone or not):

Proposition 2.3.5

$$I(f) = \sum_{S\subseteq[n]} |S|\hat{f}(S)^2.$$

Benjamini et al. [BKS99], influenced by Talagrand [Tal96], introduced the following quantity:

Definition 2.3.6

$$I\!I(f) = \sum_{j=1}^{n} I_j(f)^2.$$

Since we are interested in asymptotic analysis of noise sensitivity, for a given function f it is natural to look at the behavior of the derivatives of $\mathsf{NS}_\epsilon(f)$ (with respect to ϵ) near 0 and $\frac{1}{2}$; these quantities are related to I, $I\!I$, and f's Fourier coefficients of degree 1:

Proposition 2.3.7 *For any boolean function f, $\mathsf{NS}'_\epsilon(f)$ is a decreasing function of ϵ on $[0, \frac{1}{2}]$, with*

$$\begin{aligned}\mathsf{NS}'_0(f) &= \sum_{S\subseteq[n]} |S|\hat{f}(S)^2 = I(f) \\ &= \sum_{|S|=1} \hat{f}(S) \qquad \textit{(if } f \textit{ is monotone),}\end{aligned}$$

$$\begin{aligned}\mathsf{NS}'_{\frac{1}{2}}(f) &= \sum_{|S|=1} \hat{f}(S)^2 \\ &= I\!I(f) \qquad\qquad \textit{(if } f \textit{ is monotone).}\end{aligned}$$

Proof: $\mathsf{NS}'_\epsilon(f)$ is decreasing as a function of ϵ because $\mathsf{NS}_\epsilon(f)$ is concave (Proposition 2.1.9). The formulas for $\mathsf{NS}'_0(f)$ and $\mathsf{NS}'_{\frac{1}{2}}(f)$ follow by differentiating the formula in Proposition 2.3.1, and using Proposition 2.3.5 and the fact that $I_j(f) = \hat{f}(\{j\})$ if f is a monotone function. □

2.4 The hypercontractivity theorem and its consequences

Let us introduce a functional operator closely connected to noise sensitivity:

Definition 2.4.1 *For each $\rho \in [-1, 1]$, the* Bonami-Beckner operator *T_ρ is a linear operator that maps the space of functions $\{+1, -1\}^n \to \mathbf{R}$ into itself via*

$$T_\rho(f) = \sum_{S\subseteq[n]} \rho^{|S|}\hat{f}(S)\chi_S,$$

or equivalently,

$$T_\rho(f)(x) = \mathop{\mathbf{E}}_{y=N_{\frac{1}{2}-\frac{1}{2}\rho}(x)}[f(y)].$$

Note that the name "Bonami-Beckner operator" is not used consistently in the computer science literature. Also note that we usually only consider T_ρ for $\rho \in [0, 1]$. The equivalence between the two definitions given follows easily from the considerations in Proposition 2.3.1.

Let us list some simple properties of the Bonami-Beckner operator:

Fact 2.4.2 *Let $f\colon \{+1,-1\}^n \to \mathbf{R}$.*

- *$T_0(f) = \mathbf{E}[f]$, $T_1(f) = f$, and $T_{-1}(f)(x) = f(-x)$. Also $\mathbf{E}[T_\rho(f)] = \mathbf{E}[f]$.*
- *For all $\rho \neq 0$, T_ρ is a 1-1 operator on the space of functions $\{+1,-1\}^n \to \mathbf{R}$. (Proof: use the first definition of T_ρ.)*
- *For all $\rho \neq 0$, T_ρ is a self-adjoint operator on the space of functions $\{+1,-1\}^n \to \mathbf{R}$; i.e., $\langle T_\rho f, g\rangle = \langle f, T_\rho g\rangle$. (Proof: use the second definition and the first point of Fact 2.1.4.)*
- *For all $|\rho| < 1$, T_ρ is a* positivity-improving *operator — i.e., if $f \geq 0$ and $f \neq 0$ then $T_\rho(f) > 0$. If $|\rho| = 1$ then T_ρ is* positivity-preserving *(i.e., $T_\rho(f) \geq 0$ in the same circumstances). (Proof: use the second definition of T_ρ.)*
- *For every $p \geq 1$, T_ρ is a contraction in $L_p(\{+1,-1\}^n)$: $||T_\rho f||_p \leq ||f||_p$. For every $p \leq 1$, T_ρ is an expansion in $L_p^+(\{+1,-1\}^n)$: $||T_\rho f||_p \geq ||f||_p$ for $f \geq 0$.[1] (Proof: use the second definition of T_ρ and Jensen's inequality.)*
- *If f is a boolean function — i.e., has range $\{+1,-1\}$ — and $\epsilon \in [0, \frac{1}{2}]$, then $\mathsf{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2}||T_{\sqrt{1-2\epsilon}}(f)||_2^2$.*

The last fact above provides the connection between the Bonami-Beckner operator and noise sensitivity; studying the noise sensitivity of f is equivalent to studying the 2-norm of the Bonami-Beckner operator applied to f.

In the remainder of this subsection we discuss a very powerful theorem concerned with norms and the Bonami-Beckner operator. As we saw in Fact 2.4.2, for $p \geq 1$, T_ρ is a contractive map in L_p. In fact, the Bonami-Beckner operator is a *hypercontractive* map; i.e., it contracts as an operator from L_q to L_p for some $q > p$. The following theorem was first proved by Bonami [Bon68], in a form equivalent to the second statement of the theorem:

[1] The case $p = 0$ is a removable singularity; by $||f||_0$ we mean the geometric mean of $|f|$'s values.

Theorem 2.4.3 *Let $f\colon \{+1,-1\}^n \to \mathbf{R}$ and $q \geq p \geq 1$. Then*

$$||T_\rho(f)||_q \leq ||f||_p \qquad \text{for all } 0 \leq \rho \leq \left(\frac{p-1}{q-1}\right)^{\frac{1}{2}}.$$

Equivalently, if f has $\hat{f}(S) = 0$ for all $|S| > d$, then

$$||f||_q \leq \left(\frac{q-1}{p-1}\right)^{\frac{d}{2}} ||f||_p \qquad \text{for all } q \geq p \geq 1.$$

Theorem 2.4.3 is frequently only stated and used in the case $q = 2$.

Some words about the history of Theorem 2.4.3 are in order. Theorem 2.4.3 is usually referred to in the computer science literature as the Bonami-Beckner inequality (or even just Beckner's inequality). It was first proved by Bonami in 1968 [Bon68], though not precisely in the format stated above; see Bonami's 1970 paper [Bon70] for more explanation and an explicit proof. Bonami's paper was inspired in part by a 1960 paper of Rudin [Rud60] in which a similar theorem is proved in the setting $f\colon \mathbf{Z}_n \to \mathbf{R}$. Bonami's papers were in French and it appears they were not widely known to English-speaking researchers until the early 1990's — see [Gro92]. Kahn, Kalai, and Linial [KKL88] were the first to use Theorem 2.4.3 in computer science, and they attributed the result to Beckner [Bec75]. Beckner was motivated in part by *Nelson's inequality* [Nel73], a well-known continuous analogue of Theorem 2.4.3 for real-valued functions f in one-dimensional Gaussian space. Hypercontractive inequalities in this setting have been studied fairly intently and are closely related to logarithmic Sobolev inequalities — see [Bak92, Gro92, ABC$^+$02]. Interestingly, it is known ([Gro75]) that Nelson's inequality can be proved very simply from the Bonami-Beckner theorem using a limiting argument. There are now many known proofs of the Bonami-Beckner inequality; see [Gre02, Šte02] for particularly short ones.

Several deep theorems about noise sensitivity and influences of boolean functions are proved using the Bonami-Beckner inequality. As mentioned, Kahn et al. [KKL88] were the first to use the result in computer science. They proved the following lower bound on II, which in turn shows that every balanced boolean function has a variable with influence $I_j \geq \Omega(\frac{\log n}{n})$:

Theorem 2.4.4 *If $f\colon \{+1,-1\}^n \to \{+1,-1\}$ is a boolean function, then*

$$I\!I(f) \geq \Omega\left(\mathrm{Var}[f]^2 \frac{\log^2 n}{n}\right).$$

Talagrand [Tal94] improved this result:

Theorem 2.4.5 *If $f\colon \{+1,-1\}^n \to \{+1,-1\}$ where $\{+1,-1\}^n$ is the product space in which $\mathbf{P}[-1] = p$, then*

$$\sum_{j=1}^{n} \frac{I_j^{(p)}(f)}{\log(1/I_j^{(p)}(f))} \geq \Omega(p(1-p)\mathrm{Var}[f]).$$

In 1998 Friedgut [Fri98] used the Bonami-Beckner inequality to show that functions with very low average sensitivity (significantly smaller than $\log n$, say) are close to juntas:

Theorem 2.4.6 *For any boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ and $\delta > 0$, f is a $(\delta, 2^{O(I(f)/\delta)})$-junta.*

Bourgain's Theorem 1.2.2 which shows that functions with low noise sensitivity are close to juntas also crucially uses the Bonami-Beckner inequality, as does the related paper of Kindler and Safra [KS02, Kin02].

Finally, one of Benjamini, Kalai, and Schramm's [BKS99] main theorems was the following:

Theorem 2.4.7 *If a family of functions (f_n) satisfies $I\!I(f_n) \to 0$ then it is "asymptotically noise sensitive" in the sense that $(\mathsf{NS}_{\frac{1}{2}}(f_n) - \mathsf{NS}_\epsilon(f_n)) \to 0$ as $n \to \infty$ for every $\epsilon \in (0, \frac{1}{2})$. The converse also holds when the functions f_n are monotone.*

The proof is based on another theorem of theirs which uses the Bonami-Beckner inequality. This second theorem says that the ℓ_2 Fourier weight of monotone functions at low levels can be bounded in terms of $I\!I$. The case $k = 2$ in the theorem is due to Talagrand [Tal96]; the bounds on the constant C_k are due to Kindler [Kin03].

Theorem 2.4.8 *For each $k \geq 1$ there is a constant $C_k < \infty$ such that for all monotone boolean functions f,*

$$\sum_{|S|=k} \hat{f}(S)^2 \leq C_k I\!I(f) \ln^{k-1}(e/I\!I(f)).$$

In general, C_k may be taken to be $k^{O(k)}$. If $I\!I(f) < \exp(-k/e - 1)$ then C_k may be taken to be $O(1)/k$.

We conclude by recording two known extensions of the Bonami-Beckner inequality. In [Tal94] Talagrand showed how the inequality (with worse constants) could be extended to product measures on $\{+1, -1\}^n$ in which $\mathbf{P}[-1] = p$, generalizing the case $p = \frac{1}{2}$. In 1982 Borell [Bor82] observed that the *reverse Bonami-Beckner inequality* holds:

Theorem 2.4.9 *Let $f\colon \{+1, -1\}^n \to \mathbf{R}^{\geq 0}$ be nonnegative and let $q \leq p \leq 1$. Then*

$$||T_\rho(f)||_q \geq ||f||_p \qquad \textit{for all } 0 \leq \rho \leq \left(\frac{p-1}{q-1}\right)^{\frac{1}{2}}.$$

Equivalently, if f has $\hat{f}(S) = 0$ for all $|S| > d$, then

$$||f||_q \geq \left(\frac{q-1}{p-1}\right)^{\frac{d}{2}} ||f||_p \qquad \textit{for all } q \leq p \leq 1.$$

The reverse Bonami-Beckner inequality does not appear to have been used previously in the literature on boolean functions.

Chapter 3

The Noise Sensitivity Of Specific Functions

In this chapter we analyze the noise sensitivity of several functions and classes of functions. Since exact calculations are not possible for most functions, we will often be interested in asymptotic estimates; for example, we will be interested in showing lower and upper bounds on the noise sensitivity of functions as the noise parameter ϵ goes to 0 or $\frac{1}{2}$.

In analyzing a function f, whenever possible we shall include plots of the "noise sensitivity graph" of f and of the "Fourier spectrum graph" of f. The former is simply a plot of $\mathsf{NS}_\epsilon(f)$ as a function of $\epsilon \in [0, \frac{1}{2}]$; the latter is a plot of $\sum_{|S|=k} \hat{f}(S)^2$ versus k for $k = 0 \ldots n$.

3.1 PARITY

From Proposition 2.3.1 we immediately have the following:

Proposition 3.1.1

$$\mathsf{NS}_\epsilon(\mathsf{PARITY}_n) = \mathsf{NS}_\epsilon(-\mathsf{PARITY}_n) = \frac{1}{2} - \frac{1}{2}(1-2\epsilon)^n.$$

Noise sensitivity graphs of PARITY are shown in Figure 3-1. Of course,

$$\widehat{\mathsf{PARITY}}_n(S) = \begin{cases} 1 & \text{if } |S| = n, \\ 0 & \text{otherwise.} \end{cases}$$

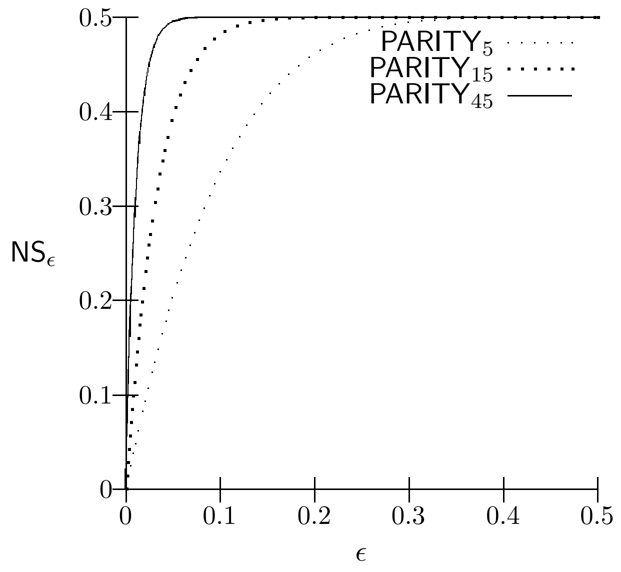

Figure 3-1: Noise sensitivity graphs of PARITY

Hence the Fourier spectrum graph of PARITY_n is as shown in Figure 3-2. From this fact, we may also deduce the well-known fact that PARITY is the most noise sensitive boolean function:

Proposition 3.1.2 *For each fixed n and $\epsilon \in (0, 1/2)$, the functions $\pm\mathsf{PARITY}_n$ are the only n-bit boolean functions of maximum NS_ϵ.*

Proof: By Proposition 2.3.1, for any boolean function on n bits $f\colon \{+1, -1\}^n \to \{+1, -1\}$,

$$\mathsf{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2} \sum_{S \subseteq [n]} (1 - 2\epsilon)^{|S|} \hat{f}(S)^2.$$

But we also know from Parseval that $\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1$. It follows immediately that any function which has all of its Fourier weight on the coefficient of degree n maximizes noise sensitivity. PARITY_n and its negation are the only such functions. □

Note that any function which is an embedded parity — i.e., is given by the parity of some subset S of its bits — has the same noise sensitivity as $\mathsf{PARITY}_{|S|}$. In particular, any dictator function π_n^i has $\mathsf{NS}_\epsilon(\pi_n^i) = \epsilon$.

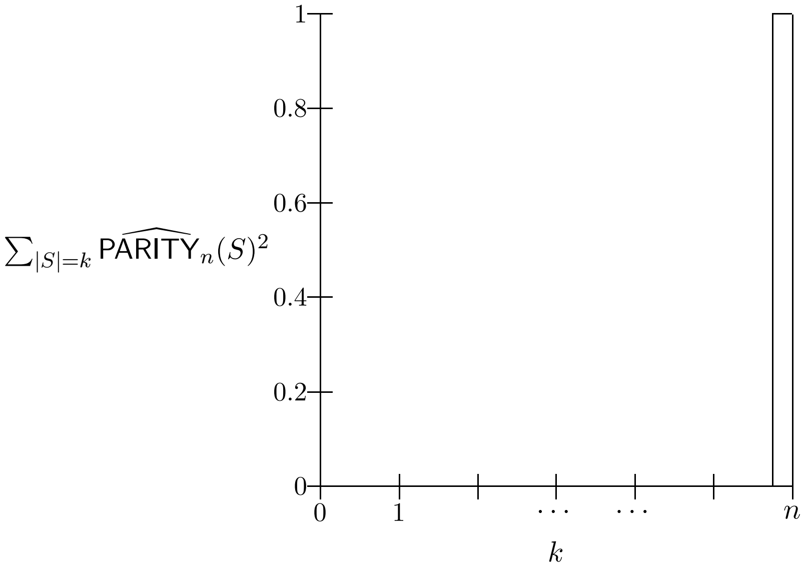

Figure 3-2: Fourier spectrum graph of PARITY_n

3.2 Dictator functions

As we saw in the previous section, any dictator function $\pm\pi_n^i$ trivially has noise sensitivity ϵ at ϵ. For completeness, we give the noise sensitivity and Fourier spectrum graphs of dictator functions in Figures 3-3 and 3-4. Since the noise sensitivity graph is concave (Proposition 2.1.9), the dictator functions are least noise sensitive among all functions with $\mathsf{NS}_{\frac{1}{2}} = \frac{1}{2}$ — i.e., all balanced functions. The dictator functions are the only functions with this property:

Proposition 3.2.1 *Fix any $\epsilon \in (0, \frac{1}{2})$. Then for all balanced $f\colon \{+1,-1\}^n \to \{+1,-1\}$, it holds that $\mathsf{NS}_\epsilon(f) \geq \epsilon$, and equality is achieved if and only if f is a dictator function.*

Proof: Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be balanced, so $\hat{f}(\emptyset) = 0$. By Proposition 2.1.9,

$$\begin{aligned}
\mathsf{NS}_\epsilon(f) &= \frac{1}{2} - \frac{1}{2}\sum_{S\neq\emptyset}(1-2\epsilon)^{|S|}\hat{f}(S)^2 \\
&\geq \frac{1}{2} - \frac{1}{2}\sum_{S\neq\emptyset}(1-2\epsilon)\hat{f}(S)^2 \\
&= \epsilon,
\end{aligned}$$

with equality if and only if f has no Fourier coefficients above degree 1. We claim this implies f is a dictator function. To prove this we show that in this case, $\hat{f}(\{i\}) \neq 0$ for at

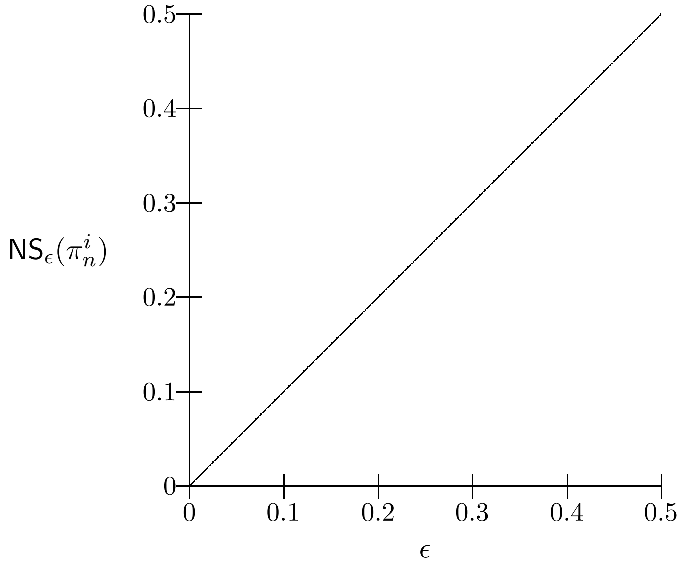

Figure 3-3: Noise sensitivity graph of π_n^i

most one i; then f depends on at most one input, as required.

Suppose to the contrary that $\hat{f}(\{i\}), \hat{f}(\{j\}) \neq 0$, for $i \neq j$. Fix any settings for the x_k's with $k \neq i, j$, and then consider x_i and x_j varying over $\{+1, -1\}$. Then $\hat{f}(\{i\})x_i + \hat{f}(\{j\})x_j$ takes on at least three distinct values; hence $f(x) = \sum_{|S|=1} \hat{f}(S)x_S$ cannot always take values in $\{+1, -1\}$, a contradiction. □

Friedgut, Kalai, and Naor [FKN01] proved a significant generalization: any function with $\sum_{|S|>1} \hat{f}(S)^2 \leq \epsilon$ is an $(O(\epsilon), 1)$-junta (i.e., is close to a dictator function or a constant). See Kalai and Safra [KS02] for further generalization.

3.3 AND and OR

Since AND and OR are the same function up to switching the roles of ±1, the noise sensitivity properties of the two functions are the same. The following proposition is immediate from Fact 2.1.7:

Proposition 3.3.1

$$\mathsf{NS}_\epsilon(\mathsf{AND}_n) = \mathsf{NS}_\epsilon(\mathsf{OR}_n) = 2^{1-n}(1 - (1-\epsilon)^n).$$

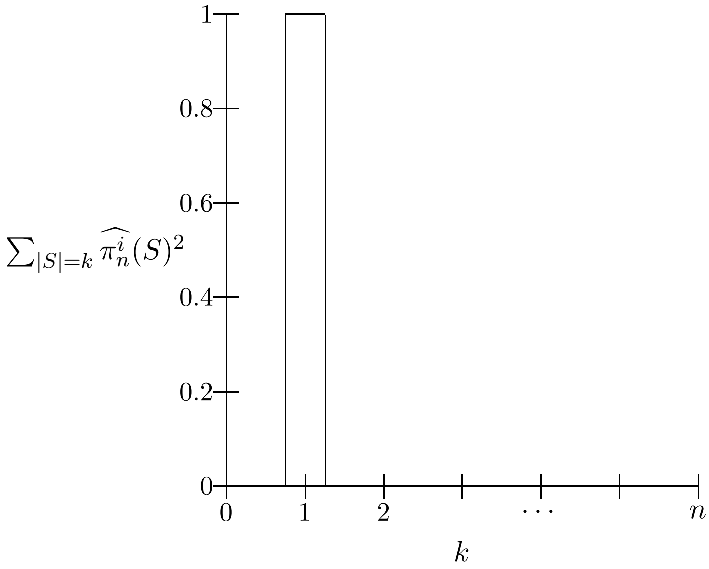

Figure 3-4: Fourier spectrum graph of π_n^i

Some noise sensitivity graphs of AND_n (equivalently, of OR_n) are shown in Figure 3-5. It easy (see, e.g., [Man95]) to check the following:

$$\widehat{\mathsf{AND}}_n(S) = \begin{cases} 1 - 2^{1-n} & \text{if } S = \emptyset, \\ (-1)^{|S|+1} 2^{1-n} & \text{otherwise.} \end{cases}$$

$$\widehat{\mathsf{OR}}_n(S) = \begin{cases} -1 + 2^{1-n} & \text{if } S = \emptyset, \\ 2^{1-n} & \text{otherwise.} \end{cases}$$

Note that all Fourier coefficients of positive degree have the same magnitude. A representative Fourier spectrum graph of AND_n (equivalently, of OR_n) is shown in Figure 3-6.

Given some boolean functions $f_1, \ldots, f_m$ we will be interested in computing the noise sensitivity of $\mathsf{AND}(f_1, \ldots, f_m)$, the function given by applying $f_1, \ldots, f_m$ on disjoint sets of variables and ANDing together the results. To emphasize the fact that the functions are on disjoint sets of variables, we call this function the *read-once* AND *(or intersection) of* $f_1, \ldots, f_m$. It is easy to see that the noise sensitivity of $\mathsf{AND}(f_1, \ldots, f_m)$ depends only on the noise sensitivity of the f_i's and the probability that the f_i's are T:

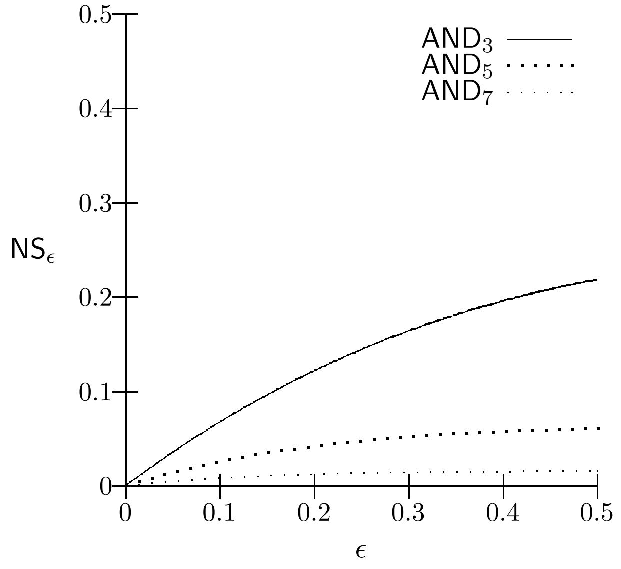

Figure 3-5: Noise sensitivity graphs of AND (equivalently, OR)

Proposition 3.3.2 *Let $f_1, \dots, f_m$ be boolean functions and write $p_i = \mathbf{P}[f(x) = \mathrm{T}]$, $\eta_i = \mathsf{NS}_\epsilon(f_i)$. Let $g = \mathsf{AND}(f_1, \dots, f_m)$. Then*

$$\mathsf{NS}_\epsilon(g) = 2\left(\prod_{i=1}^{m} p_i\right)\left(1 - \prod_{i=1}^{m}(1 - \eta_i/2p_i)\right).$$

Proof: Using Fact 2.1.7 and the independence we get from the fact that the f_i's are defined on disjoint sets of variables, we have

$$\begin{aligned}\mathsf{NS}_\epsilon(g) &= 2\,\mathbf{P}[g(x) = \mathrm{T}]\,\mathbf{P}[g(N_\epsilon(x)) = \mathrm{F} \mid g(x) = \mathrm{T}] \\ &= 2\left(\prod_{i=1}^{m} \mathbf{P}[f_i(x) = \mathrm{T}]\right)\left(1 - \prod_{i=1}^{m} \mathbf{P}[f_i(N_\epsilon(x)) = \mathrm{T} \mid f_i(x) = \mathrm{T}]\right) \\ &= 2\left(\prod_{i=1}^{m} p_i\right)\left(1 - \prod_{i=1}^{m}(1 - \eta_i/2p_i)\right),\end{aligned}$$

where the last line uses Fact 2.1.7 again. □

3.4 Majority

As there is a closed form for the Fourier coefficients of MAJ_n, many questions about the noise sensitivity of majority can be answered by explicit calculation. However we shall begin

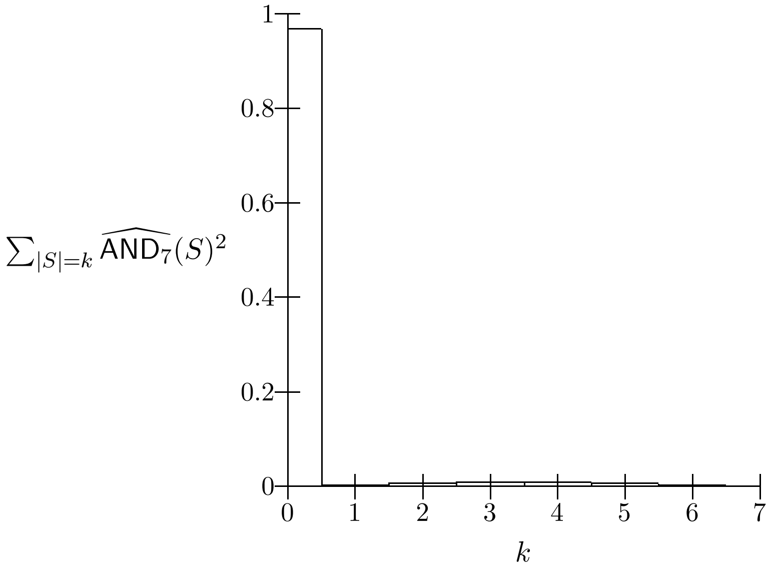

Figure 3-6: Fourier spectrum graph of AND_7

by taking a more qualitative route. First, a lemma which describes the sums of the bits in $(x, N_\epsilon(x))$ in the limit as $n \to \infty$:

Proposition 3.4.1 *Let $x \in \{+1,-1\}^n$ be chosen uniformly at random, and let $y = N_\epsilon(x)$, where $\epsilon \in [0,1]$. Write $\rho = 1-2\epsilon$. Let $X = n^{-\frac{1}{2}} \sum_{i=1}^n x_i$ and $Y = n^{-\frac{1}{2}} \sum_{i=1}^n y_i$. Then as $n \to \infty$, the pair of random variables (X,Y) approaches the jointly normal distribution ϕ_Σ with 0 means and covariance matrix $\Sigma = \Sigma(\rho) = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. As an error bound, we have that for any convex region $R \subseteq \mathbf{R}^2$, $|\mathbf{P}[(X,Y) \in R] - \phi_\Sigma(R)| \leq O((1-\rho^2)^{-\frac{1}{2}} n^{-\frac{1}{2}})$.*

Proof: This follows from the Central Limit Theorem (see, e.g., [Fel68]), noting that for each coordinate i, $\mathbf{E}[x_i^2] = \mathbf{E}[y_i^2] = 1$, $\mathbf{E}[x_i y_i] = \rho$. The Berry-Esséen-type error bound is proved in Sazonov [Saz81, p. 10 item 6]. □

Theorem 3.4.2 *For any $\epsilon \in [0,1]$,*

$$\lim_{n\to\infty} \mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = \frac{1}{2} - \frac{1}{\pi}\arcsin(1-2\epsilon),$$

with the following error estimate:

$$\left|\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) - \left(\frac{1}{2} - \frac{1}{\pi}\arcsin(1-2\epsilon)\right)\right| \leq O((\epsilon(1-\epsilon)n)^{-\frac{1}{2}}). \tag{3.1}$$

Indeed, for $\epsilon \in [0, \frac{1}{2}]$*,* $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n)$ *in increasing in* n*, and for* $\epsilon \in [\frac{1}{2}, 1]$*,* $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n)$ *is decreasing in* n*, so the error estimate (3.1) is always one-sided.*

Proof: Using the notation from Proposition 3.4.1, by definition we have $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = \mathbf{P}[(X,Y) \in R]$, where R is the union of the upper-left and lower-right quadrants of $\mathbf{R}^2$, $R = Q_{+-} \cup Q_{-+}$. But each of Q_{+-} and Q_{-+} is convex and is well known to have probability mass $\frac{1}{4} - \frac{1}{2\pi}\arcsin\rho$ under ϕ_Σ (see, e.g., [AS72, 26.3.19]). This completes the proof of all of Theorem 3.4.2 except for the claim that $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n)$ increases with n for $\epsilon \in [0, \frac{1}{2}]$ and decreases with n for $\epsilon \in [\frac{1}{2}, 1]$.

The proof of this claim uses the Fourier calculations of Theorem 3.5.2; in particular, the fact that $M_n(k) = \sum_{|S|=k} \widehat{\mathsf{MAJ}}_n(S)^2$ decreases in n for all k, and is constantly 0 for even k. We conclude from Proposition 2.3.1,

$$\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = \frac{1}{2} - \frac{1}{2}\sum_{k \text{ odd}} (1-2\epsilon)^k M_n(k).$$

We know that $\sum_k M_n(k) = 1$ for every n, but as n increases, these nonnegative weights “shift up” and put more weight at higher levels k. Now simply note that $(1-2\epsilon)^k$ is decreasing in k for $\epsilon \leq \frac{1}{2}$ and increasing in (odd) k for $\epsilon \geq \frac{1}{2}$. □

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

To make a basic approximation, recall ([AS72, 4.4.41]) the following:

Fact 3.4.3 $\arcsin(1-x) = \frac{1}{2} - (2x)^{\frac{1}{2}} - O(x^{\frac{3}{2}})$.

Thus from Theorem 3.4.2 we conclude:

Corollary 3.4.4 *For every* n*,*

$$\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) \leq ((2/\pi) + o(1))\sqrt{\epsilon},$$

where the $o(1)$ *refers to* $\epsilon \to 0$*.*

Note that in the range $\epsilon \in [\Omega(1), \frac{1}{2}]$, the error bound (3.1) is $O(n^{-\frac{1}{2}})$ so the approximation $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = \frac{1}{2} - \frac{1}{\pi}\arcsin(1-2\epsilon)$ is very good. Indeed, even for ϵ as small as $\Omega(n^{-\frac{1}{2}})$ the error term is no bigger than the actual limiting value; using Fact 3.4.3 again we get,

Corollary 3.4.5 *There is a constant $C < \infty$ such that*

$$\mathsf{NS}_{Cn^{-\frac{1}{2}}}(\mathsf{MAJ}_n) = \Theta(n^{-\frac{1}{4}}).$$

For $\epsilon = o(n^{-\frac{1}{2}})$, the error bound in Theorem 3.4.2 breaks down and we need to use alternative methods. First, we can find $\mathsf{NS}'_0(\mathsf{MAJ}_n)$: Since MAJ_n is monotone, Proposition 2.3.7 lets us compute $\mathsf{NS}'_0(\mathsf{MAJ}_n)$ (and $\mathsf{NS}'_{\frac{1}{2}}(\mathsf{MAJ}_n)$) in terms of the degree-one Fourier coefficients of MAJ_n. It is easy to check that $\widehat{\mathsf{MAJ}_n}(\{j\}) = \binom{n-1}{\frac{n-1}{2}}2^{1-n}$ for all $j \in [n]$. Hence, by Proposition 2.3.7 and Stirling's approximation, we have the following:

Proposition 3.4.6

- $\mathsf{NS}'_0(\mathsf{MAJ}_n) = I(\mathsf{MAJ}_n) = n2^{1-n}\binom{n-1}{\frac{n-1}{2}} = ((2/\pi)^{\frac{1}{2}} + o(1))n^{\frac{1}{2}}$,
- $\mathsf{NS}'_{\frac{1}{2}}(\mathsf{MAJ}_n) = I\!I(\mathsf{MAJ}_n) = n2^{2-2n}\binom{n-1}{\frac{n-1}{2}} = \frac{2}{\pi} + o(1)$.

For $\epsilon \le n^{-1}$ the following simple estimate proves useful:

Proposition 3.4.7 *For $n \ge 3$ and $\epsilon \le n^{-1}$,*

$$\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) \ge (2/\pi)^{\frac{1}{2}}\, n^{\frac{1}{2}}\, \epsilon\, \exp(-1/3n)\exp(-\epsilon n).$$

Proof: Suppose x is chosen uniformly at random and y is formed from x by flipping each bit independently with probability ϵ. Then

$$\begin{aligned}\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) &= \mathbf{P}[\mathsf{MAJ}_n(x) \ne \mathsf{MAJ}_n(y)] \\ &\ge \mathbf{P}[\mathsf{MAJ}k(x) \ne \mathsf{MAJ}k(y) \mid \text{exactly one flip}] \times \mathbf{P}[\text{exactly one flip}], \qquad (3.2)\end{aligned}$$

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

$$\mathbf{P}[\text{exactly one flip}] \ge k\delta\exp(-\delta k). \qquad (3.3)$$

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.,

$$\begin{aligned}\mathbf{P}[\mathsf{MAJ}_n(x) \neq \mathsf{MAJ}_n(y) \mid \text{exactly one flip}] &= \binom{n-1}{\frac{n-1}{2}} 2^{1-n} \\ &\geq (2/\pi)^{\frac{1}{2}}\, n^{-\frac{1}{2}}(1-1/4n) \geq (2/\pi)^{\frac{1}{2}}\, n^{\frac{1}{2}} \exp(-1/3n), \qquad (3.4)\end{aligned}$$

where the first inequality follows by Stirling's formula and the second since $1 - 1/4n \leq \exp(-1/3n)$ for $n \geq 3$. Combining (3.2), (3.3) and (3.4) we obtain the required result. □

By combining Propositions 3.4.6 and 3.4.7 we get the following:

Proposition 3.4.8 *For $\epsilon = o(n^{-1})$, $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = ((2/\pi)^{\frac{1}{2}} + o(1))n^{\frac{1}{2}}\ \epsilon$. For $\epsilon = \Theta(n^{-1})$, $\mathsf{NS}_\epsilon(\mathsf{MAJ}_n) = \Theta(n^{-\frac{1}{2}})$.*

Proof: To get the upper bounds, use Proposition 3.4.7 and standard estimates. To get the lower bounds, use Proposition 3.4.6 and the Intermediate Value Theorem, noting that $\mathsf{NS}'_\epsilon(f)$ is decreasing in ϵ and hence maximized at $\epsilon = 0$. □

Theorem 3.4.2, Corollary 3.4.5, and Propositions 3.4.6 and 3.4.8 give a fairly complete picture of the noise sensitivity graph of MAJ_n. Some noise sensitivity graphs of MAJ are shown in Figure 3-7.

3.5 The Fourier spectrum of majority

In this section we will examine the Fourier spectrum graph of MAJ_n. Recall the Taylor expansion of $\arcsin x$ ([AS72, 4.4.40]):

Fact 3.5.1

$$\begin{aligned}\arcsin x &= x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \frac{5}{112}x^7 + \cdots \\ &= \sum_{k\ odd} k^{-1}\binom{k-1}{\frac{k-1}{2}} 2^{1-k}\, x^k.\end{aligned}$$

We shall write $[x^k](\arcsin x)$ for the degree k coefficient in the above,

$$[x^k](\arcsin x) = \begin{cases} 0 & \text{if } k \text{ is even,} \\ k^{-1}\binom{k-1}{\frac{k-1}{2}} 2^{1-k} & \text{if } k \text{ is odd.}\end{cases}$$

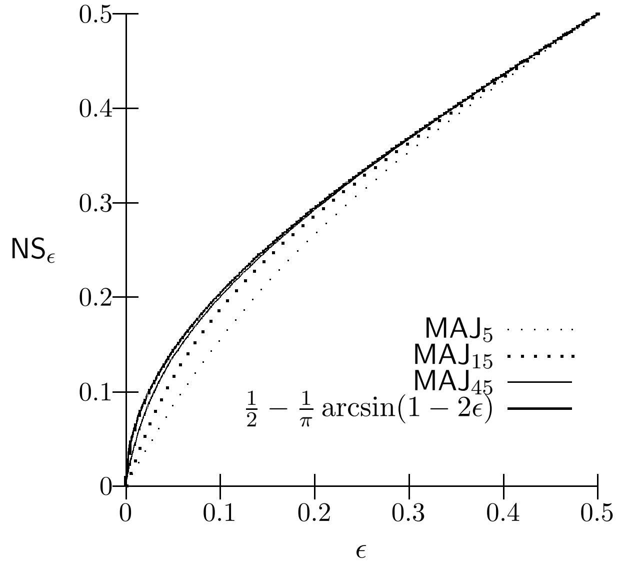

Figure 3-7: Noise sensitivity graphs of MAJ

By comparing Theorem 3.4.2 with Proposition 2.3.1, we expect that $\sum_{|S|=k} \widehat{\mathsf{MAJ}}_n(S)^2$ should go to $[x^k]((2/\pi)\arcsin x)$ as $n \to \infty$, and this is indeed true:

Theorem 3.5.2 *For odd n and $0 \leq k \leq n$, write $M_n(k) = \sum_{|S|=k} \widehat{\mathsf{MAJ}}_n(S)^2$. Then for each fixed $k \in \mathbf{N}$,*

$$\lim_{n\to\infty} M_n(k) = [x^k]((2/\pi)\arcsin x).$$

Further, $M_n(k)$ is decreasing in n for every k. Finally, for k varying with n, we have the following error estimate which holds for any $k \leq n/2$:

$$M_n(k) \leq \Big([x^k]((2/\pi)\arcsin x)\Big)\,(1 + 2k/n).$$

Proof: We begin with Bernasconi's determination of the Fourier coefficients of MAJ_n from [Ber98]:

Proposition 3.5.3 *Let $S \subseteq [n]$ satisfy $|S| = k$. Then*

$$\widehat{\mathsf{MAJ}}_n(S) = \begin{cases} (-1)^{\frac{k-1}{2}} \frac{2}{2^n} \frac{\binom{k-1}{\frac{k-1}{2}}\binom{n-k}{\frac{n-k}{2}}}{\binom{\frac{n-1}{2}}{\frac{k-1}{2}}} & \text{if } k \text{ is odd,} \\ 0 & \text{if } k \text{ is even.} \end{cases}$$

Thus our desired result holds for even k. For odd k, we get

$$M_n(k) = \frac{4}{2^{2n}} \binom{n}{k} \frac{\binom{k-1}{\frac{k-1}{2}}^2 \binom{n-k}{\frac{n-k}{2}}^2}{\binom{\frac{n-1}{2}}{\frac{k-1}{2}}^2}. \tag{3.5}$$

By explicit expansion and simplification, we get

$$\frac{M_{n+2}(k)}{M_n(k)} = \frac{(n+2)(n-k+1)}{(n+1)(n-k+2)} \leq 1,$$

and it follows that $M_n(k)$ is decreasing in n for every k, as required. It remains to compare the limiting value of (3.5) with $[x^k]((2/\pi)\arcsin x)$. Again, by explicit expansion to factorials and simplification, we get

$$\frac{M_n(k)}{[x^k]((2/\pi)\arcsin x)} = (\pi/2)n2^{k+1-2n} \binom{n-1}{\frac{n-1}{2}} \binom{n-k}{\frac{n-k}{2}}. \tag{3.6}$$

By Stirling's approximation, $\binom{m}{m/2} \nearrow (2/\pi)^{\frac{1}{2}} m^{-\frac{1}{2}} 2^m$, in the sense that the ratio of the right-hand side to the left-hand decreases to 1 as $m \to \infty$. Substituting this into (3.6) and using the same notation, we have

$$\frac{M_n(k)}{[x^k]((2/\pi)\arcsin x)} \nearrow \frac{n}{(n-1)^{\frac{1}{2}}(n-k)^{\frac{1}{2}}}. \tag{3.7}$$

When k is fixed, the right-hand side of (3.7) goes to 1 as $n \to \infty$ which proves that $M_n(k) \to [x^k]((2/\pi)\arcsin x)$ for all fixed k. In general, we have

$$\frac{n}{(n-1)^{\frac{1}{2}}(n-k)^{\frac{1}{2}}} = \left(1 - \frac{k+1}{n} + \frac{k}{n}\right)^{-\frac{1}{2}} \leq 1 + 2k/n,$$

where it can be checked that the inequality holds for all $k \leq n/2$. □

Theorem 3.5.2 gives us very good estimates for the Fourier spectrum of MAJ_n up to degree $n/2$. For higher degree — especially degree $n - O(1)$ — we can instead use the following:

Proposition 3.5.4 *For all $0 \leq k < n$,*

$$\sum_{|S|=n-k} \widehat{\mathsf{MAJ}}_n(S)^2 = \frac{k+1}{n-k} \sum_{|S|=k+1} \widehat{\mathsf{MAJ}}_n(S)^2$$

Proof: This follows easily from Proposition 3.5.3, which shows that all Fourier coefficients at degree $n-k$ have the same value as all Fourier coefficients at degree $k+1$. The quantity $(k+1)/(n-k)$ is the ratio $\binom{n}{n-k}/\binom{n}{k+1}$. □

For example, since the Fourier weight of MAJ_n at degree 1 approaches $2/\pi$, the Fourier weight at degree n approaches $2/\pi n$. Similarly, we can see that the Fourier weight at degree $n-2$ approaches $1/\pi(n-2)$.

To complete our picture of the Fourier spectrum of majority, we consider $[x^k]((2/\pi)\arcsin x)$ asymptotically in k:

Fact 3.5.5 *For all odd k,*

$$[x^k]((2/\pi)\arcsin x) > (2/\pi k)^{\frac{3}{2}},$$

with the ratio of the two quantities decreasing to 1 as $k \to \infty$ (through the odd numbers).

This follows immediately from Stirling's approximation.

Using Theorem 3.5.2 and Proposition 3.5.4 we can completely infer the Fourier spectrum graph for MAJ. A representative example is shown in Figure 3-8.

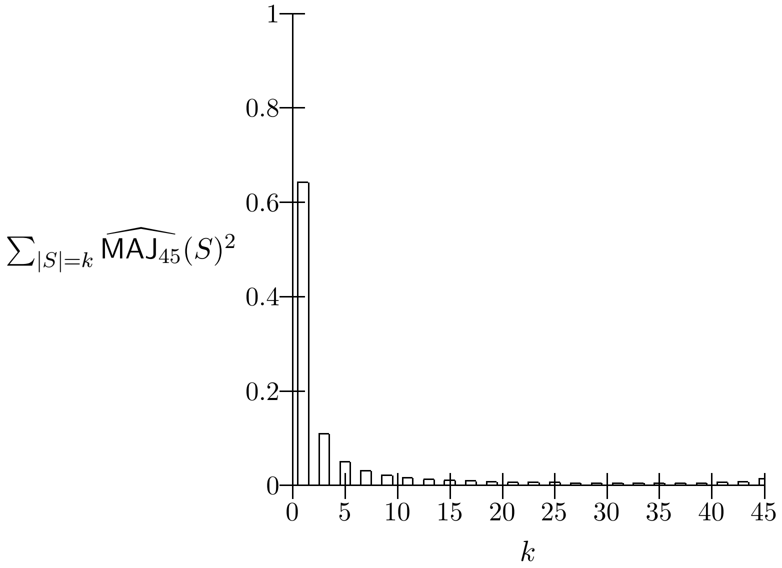

Figure 3-8: Fourier spectrum graph of MAJ_{45}

We end this section by obtaining a lower bound on the Fourier tail of majority.

Theorem 3.5.6 *For any $\epsilon = \epsilon(n) \geq n^{-\frac{1}{3}}$ the following inequality holds for all sufficiently large odd n:*

$$\sum_{|S| \geq \alpha} \widehat{\mathsf{MAJ}}_n(S)^2 > \epsilon,$$

where α is any odd number no bigger than $(1/7)\epsilon^{-2}$.

Proof: Fix ϵ, n and α as described. As in Theorem 3.5.2, write $M_n(k) = \sum_{|S|=k} \widehat{\mathsf{MAJ}}_n(S)^2$. From the same theorem, we know that for all $k \leq \alpha$,

$$M_n(k) \leq \Big([x^k]((2/\pi)\arcsin x)\Big)\,(1 + 2\alpha/n). \tag{3.8}$$

Since the Taylor series for $\arcsin x$ converges at $x = 1$ and $\frac{2}{\pi}\arcsin 1 = 1$, we conclude that $\sum_{k \text{ odd}}[x^k]((2/\pi)\arcsin x) = 1$. Hence

$$\begin{aligned}
\sum_{k<\alpha}[x^k]((2/\pi)\arcsin x) &= 1 - \sum_{\substack{k\geq\alpha \\ k \text{ odd}}} [x^k]((2/\pi)\arcsin x) \\
&< 1 - \sum_{\substack{k\geq\alpha \\ k \text{ odd}}} (2/\pi k)^{\frac{3}{2}} \qquad\qquad \text{(Fact (3.5.5))} \\
&\leq 1 - \frac{1}{2}\int_{\alpha}^{\infty} (2/\pi k)^{\frac{3}{2}}\, dk \\
&= 1 - \frac{1}{2}\left[(2/\pi)^{\frac{3}{2}} \cdot 2k^{-\frac{1}{2}}\right]_{k=\alpha} \\
&\leq 1 - (2/\pi)^{\frac{3}{2}}\alpha^{-\frac{1}{2}}.
\end{aligned} \tag{3.9}$$

Combining Equations (3.8) and (3.9), we get

$$\begin{aligned}
\sum_{k<\alpha} M_n(k) &< (1 + 2\alpha/n)(1 - (2/\pi)^{\frac{3}{2}}\alpha^{-\frac{1}{2}}) \\
&\leq 1 + 2\alpha/n - (2/\pi)^{\frac{3}{2}}\alpha^{-\frac{1}{2}} \\
&\leq 1 + (2/7)(\epsilon^2 n)^{-1} - (2/\pi)^{\frac{3}{2}}(1/7\epsilon^2)^{-\frac{1}{2}} \qquad \text{(by definition of } \alpha) \\
&\leq 1 + (2/7)\epsilon - (2/\pi)^{\frac{3}{2}}7^{\frac{1}{2}}\epsilon \qquad \text{(as } n \geq \epsilon^{-3}) \\
&< 1 - \epsilon.
\end{aligned}$$

But $\sum_{k=0}^{n} M_n(k) = 1$, so the proof is complete. □

3.6 Noise sensitivity of monotone functions

There are several reasons why it is interesting and natural to study the noise sensitivity of monotone functions. The first is that in some sense they are the "least noise sensitive" functions. Combinatorial shifting decreases not just average sensitivity but noise sensitivity as well. Recall the *(down-)shifting* operators (first introduced by Kleitman [Kle66]; see Frankl [Fra87] for a survey on shifting):

Definition 3.6.1 *Given $f\colon \{+1,-1\}^n \to \{+1,-1\}$ and $i \in [n]$, the operator κ_j is defined by $\kappa_j f\colon \{+1,-1\}^n \to \{+1,-1\}$,*

$$(\kappa_j f)(x) = \begin{cases} f(x) & \text{if } f(x) = f(\sigma_j x), \\ x_j & \text{if } f(x) \neq f(\sigma_j x). \end{cases}$$

It is well known that for any f, $\mathbf{E}[\kappa_j f] = \mathbf{E}[f]$, the function $\kappa_1\kappa_2\cdots\kappa_n f$ is monotone, and that $I_i(\kappa_j f) \leq I_i(f)$ for all i, j (this fact is proved in, e.g., [BL90]). In [BKS99] Benjamini, Kalai, and Schramm showed the following:

Proposition 3.6.2 *For every $\epsilon \in [0, \frac{1}{2}]$, $\mathsf{NS}_\epsilon(\kappa_j f) \leq \mathsf{NS}_\epsilon(f)$.*

Thus the monotone-shifted version of any function f is less noise-sensitive than f itself. Indeed, we will show in Chapter 6 that the "left-monotone-shifted" version of f is less noise-sensitive than f itself; see Proposition 6.6.4.

Another reason for studying the noise sensitivity of monotone functions is that there are some well known upper bounds on how noise sensitive monotone functions can be. In Sections 3.7, 3.8, and 3.9 we will give families of functions which are tight or close to tight for these bounds. These highly noise sensitive monotone functions will have very useful application in our study of hardness amplification within NP in Chapter 4.

Finally, monotone functions play an important role in the study of the boolean hypercube and the influences of variables on boolean functions; see, e.g., [KKL88, Tal94, Tal96, FK96, Tal97, Bou99, BKS99, MO02b].

There are several different ways to formulate the question, "How noise sensitive can a monotone function be?" A simple folklore isoperimetric inequality (whose proof does not require the use of combinatorial shifting; see, e.g., [FK96]) implies that among all monotone (indeed, unate) functions on n inputs, the majority function MAJ_n (or something like it for even n) has maximal average sensitivity. Since its average sensitivity is easily computed to be

$$I(\mathsf{MAJ}_n) = ((2/\pi)^{\frac{1}{2}} + o(1))n^{\frac{1}{2}},$$

we get that for all monotone (even unate) functions f on n inputs,

$$I(f) \leq ((2/\pi)^{\frac{1}{2}} + o(1))n^{\frac{1}{2}}. \tag{3.10}$$

This bound can be used to get the following upper bound on the noise sensitivity of monotone functions, which is considered folklore:

Proposition 3.6.3 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be monotone (or unate). Then*

$$\mathsf{NS}_\epsilon(f) \leq \frac{1}{2} - \frac{1}{2}(1-2\epsilon)^{I(\mathsf{MAJ}_n)} = \frac{1}{2} - \frac{1}{2}(1-2\epsilon)^{((2/\pi)^{\frac{1}{2}}+o(1))n^{\frac{1}{2}}}.$$

Therefore if $\mathsf{NS}_\epsilon(f) \geq \eta$, then

$$\epsilon \geq \eta/I(\mathsf{MAJ}_n) \geq ((\pi/2)^{\frac{1}{2}} - o(1))n^{-\frac{1}{2}}.$$

Proof: The second statement follows straightforwardly from the first one, which we now prove. By Parseval's identity we have $\sum_S \hat{f}(S)^2 = 1$, and by (3.10) and Proposition 2.3.5 we have $\sum_S |S|\hat{f}(S)^2 \leq I(\mathsf{MAJ}_n)$. We want to bound $\mathsf{NS}_\epsilon(f) = \frac{1}{2} - \frac{1}{2}\sum_S (1-2\epsilon)^{|S|}\hat{f}(S)^2$. Clearly a bound is the solution to the following linear programming problem:

$$\begin{array}{llll} \text{Maximize } \frac{1}{2} - \frac{1}{2}\sum_{k=0}^n (1-2\epsilon)^k x_k & & & \\ \text{subject to: } \sum_{k=0}^n x_k = 1, & \sum_{k=0}^n kx_k \leq I(\mathsf{MAJ}_n), & 0 \leq x_k \leq 1 \text{ for all } k. \end{array} \tag{3.11}$$

Since the optimum of a linear program occurs at a vertex, an optimal solution is of the form $x_m = 1$ and $x_i = 0$ for all $i \neq m$, for some m. The constraint $\sum_{k=0}^n kx_k \leq I(\mathsf{MAJ}_n)$ implies that $m \leq I(\mathsf{MAJ}_n)$. Therefore the value of $\frac{1}{2} - \frac{1}{2}\sum_{k=0}^n (1-2\epsilon)^k x_k$ is at most $\frac{1}{2} - \frac{1}{2}(1-2\epsilon)^{I(\mathsf{MAJ}_n)}$, as claimed. □

A natural goal which we will pursue in Sections 3.7 and 3.8 is to find a monotone function f on n bits such that $\mathsf{NS}_\epsilon(f) \geq \Omega(1)$ for the smallest possible quantity ϵ. This problem was implicitly posed in Benjamini et al.'s work [BKS99], in which it was noted that the iterated majority-of-3 function $\mathsf{MAJ}_3^{\otimes \log_3 n}$, studied first by Ben-Or and Linial [BL90], has at least constant noise sensitivity for ϵ as small as $O(n^{-\log_3 \frac{3}{2}}) = O(n^{-.369...})$. As an immediate corollary of Proposition 3.6.3 we have the following:

Corollary 3.6.4 *If f is monotone and $\mathsf{NS}_\epsilon(f) \geq \Omega(1)$, then $\epsilon \geq \Omega(n^{-\frac{1}{2}})$.*

Benjamini et al. conjectured that Corollary 3.6.4 was not sharp, and that the negative exponent $\frac{1}{2}$ could be replaced by some $\beta < \frac{1}{2}$. In the following sections we will see that this is *not* true. In Section 3.7 we study the recursive majority of k function for larger values of k: 5, 7, 9, etc. Previous techniques had suggested that these functions might be less sensitive to small noise than iterated MAJ_3, but this is not the case. We will see that as k grows, the recursive majority of k is sensitive to smaller and smaller noise rates:

Theorem *For every constant $\alpha < \frac{1}{2}$ and $\delta > 0$, there exists an odd $k \geq 3$ such that for ℓ sufficiently large, we have (writing $n = k^\ell$) the following:*

$$\mathsf{NS}_{n^{-\alpha}}(\mathsf{MAJ}_k^{\otimes \ell}) \geq \frac{1}{2} - \delta.$$

(See Theorem 3.7.2 for more details.)

Thus iterated MAJ_k functions are nearly as sensitive to low noise as is possible for monotone functions. In the same section we shall further give an explicit infinite family of monotone boolean functions which have constant noise sensitivity at $\epsilon = n^{-\frac{1}{2}} \log^{O(1)} n$:

Theorem 3.7.4 *For every sufficiently small $\epsilon > 0$,*

$$\mathsf{NS}_{\epsilon/M}(\mathsf{EM}_n) \geq \epsilon - O(\epsilon^2),$$

where

$$M = n^{\frac{1}{2}}/\Theta(\log^t n), \qquad t = \log_2 \sqrt{\pi/2} = .3257\ldots.$$

Here the function family (EM_n) is also a family of iterated majorities, but these majorities are of varying, nonconstant arity. Finally, in Section 3.8 we will analyze a random family of monotone functions introduced by Talagrand in [Tal96] and show (non-constructively) the existence of one with constant noise sensitivity at $\epsilon = O(n^{-\frac{1}{2}})$:

Theorem 3.8.1 *Talagrand's construction shows there exists an infinite family of monotone functions* (C_n) *satisfying*

$$\mathsf{NS}_{n^{-\frac{1}{2}}}(C_n) \geq \Omega(1).$$

This shows that Corollary 3.6.4 is tight up to constant factors.

At the other end of the spectrum we naturally have the following question: As a function of n, how close can $\mathsf{NS}_{\frac{1}{2}-\Omega(1)}(f)$ be to $\frac{1}{2}$ if f is monotone? Kahn, Kalai, and Linial's theorem provides a bound as an immediate corollary: Since every monotone function f with $\mathsf{NS}_{\frac{1}{2}}(f) \geq \frac{1}{3}$ satisfies $\mathsf{NS}'_{\frac{1}{2}}(f) = \mathit{II}(f) \geq \Omega(\frac{\log^2 n}{n})$ (Theorem 2.4.4 and Proposition 2.3.7), and since $\mathsf{NS}_\epsilon(f)$ is an increasing concave function of ϵ (Proposition 2.1.9), we get the following:

Corollary 3.6.5 *If* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is monotone (or unate), then*

$$\mathsf{NS}_{\frac{1}{2}-\delta}(f) \leq \frac{1}{2} - \Omega\left(\frac{\log^2 n}{n}\right)\delta.$$

Ben-Or and Linial introduced a family of monotone functions in [BL90] called the *tribes* functions; we define this family in Section 3.9. Kahn et al. showed that the tribes functions were tight for their theorem, Theorem 2.4.4; however, this does not prove that that tribes functions are tight for Corollary 3.6.5. (For example, although MAJ_n has maximal NS'_0 among monotone functions (up to a constant), it is nowhere close to being tight for Corollary 3.6.4.) In Section 3.9 we exactly calculate the noise sensitivity of the tribes functions and show that they are indeed close to sharp for Corollary 3.6.5; for $\delta \leq \frac{1}{\log n}$ they are sharp up to a constant factor:

Corollary 3.9.7 *If* $\delta \leq O(\frac{1}{\log n})$ *then* $\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \geq \frac{1}{2} - O(\frac{\log^2 n}{n}\ \delta)$.

For $\delta \leq \frac{1}{2} - \Omega(1)$, combining the tribes functions with iterated majorities lets us get a family of monotone functions sharp for Corollary 3.6.5 up to a logarithmic factor:

Theorem 3.9.8 *Let* $\epsilon \geq \Omega(1)$. *Then there is an infinite family of monotone functions* (T_n) *satisfying*

$$\mathsf{NS}_\epsilon(T_n) \geq \frac{1}{2} - \frac{\log^{1+u'} n}{n},$$

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

3.7 Iterating majority and other functions

Suppose we have a balanced function on a constant number of bits, $f\colon \{+1,-1\}^k \to \{+1,-1\}$. By using the composition operator $\otimes$, we get a natural family of functions $(f^{\otimes^\ell})_\ell$ defined for infinitely many input lengths k, k^2, k^3, etc. The noise sensitivity of this family of functions is inherited from that of f in a simple way: Let $p(\epsilon) = \mathsf{NS}_\epsilon(f)$; then by Proposition 2.2.7, $\mathsf{NS}_\epsilon(f^{\otimes^\ell}) = p^{(\ell)}(\epsilon) = \underbrace{p(p(\cdots p(\epsilon)))}_{\ell \text{ times}}$. Note that p is a concave function on $[0, \frac{1}{2}]$ which is 0 at 0 and $\frac{1}{2}$ at $\frac{1}{2}$. If we begin iterating p at a very small value ϵ then the speed at which the result approaches $\Omega(1)$ is governed by $p'(0)$. Similarly, if we start iterating p at $\frac{1}{2} - \Omega(1)$, the speed at which the result approaches $\frac{1}{2}$ is governed by $p'(\frac{1}{2})$. We have the following result:

Proposition 3.7.1 *Let $f\colon \{+1,-1\}^k \to \{+1,-1\}$ be a balanced function, let $a = \mathsf{NS}'_0(f)$, $b = \mathsf{NS}'_{\frac{1}{2}}(f)$, and assume $a > 1$, $b < 1$. Then $\mathsf{NS}_\epsilon(f^{\otimes^\ell}) \geq \frac{1}{2} - \delta$ for*

$$\ell \geq \Big(\log_a(1/\epsilon) + \log_{1/b}(1/\delta)\Big)\,(1 + r(\epsilon,\delta)),$$

where $r(\epsilon, \delta) \to 0$ as $\epsilon, \delta \to 0$.

Proof: (For this proof, when a function q depends on ϵ and δ we will write $q = O(1)$ if q is uniformly bounded and we will write $q = o(1)$ if $q(\epsilon, \delta) \to 0$ as $\epsilon \to 0$, $\delta \to 0$.)

Let $p(\eta) = \mathsf{NS}_\eta(f)$. By Proposition 2.1.9 p is an increasing concave polynomial which is 0 at 0 and $\frac{1}{2}$ at $\frac{1}{2}$. Since $p'(0) = a$, $p'(\frac{1}{2}) = b$, we have $p(\eta) \geq a\eta - c\eta^2$ and $p(\frac{1}{2} - \eta) \geq \frac{1}{2} - b\eta - c'\eta^2$ for some constants c, c' which depend only on f. Recall that by Proposition 2.2.7, $\mathsf{NS}_\epsilon(f^{\otimes^\ell}) = p^{(\ell)}(\epsilon)$. We shall analyze this iteration in five steps.

- Let $\epsilon' = 1/\ln(1/\epsilon)$. As $a > 1$, we may assume without loss of generality that $a - c\epsilon' > 1$. Now whenever $\eta < \epsilon'$ we have $p(\eta) \geq (a - c\epsilon')\eta$. Thus $p^{(k)}(\eta) \geq \min\{(a - c\epsilon')^k\eta, \epsilon'\}$ for all k, and in particular, $p^{(k)}(\epsilon) \geq \epsilon'$ so long as k is at least

$$\frac{\ln(\epsilon'/\epsilon)}{\ln(a - c\epsilon')} \leq \frac{\ln(1/\epsilon)}{(1 - O(\epsilon'))\ln a} \leq (1 + o(\epsilon))\log_a(1/\epsilon).$$

- Let $r = (a-1)/2c > 0$, so $a - cr > 1$. By a similar argument to the above, we conclude that $p^{(k)}(\epsilon') \geq r$ so long as k is at least
$$\frac{\ln(r/\epsilon')}{\ln(a-cr)} \leq O(\ln(1/\epsilon')) \leq o(\log_a(1/\epsilon)).$$

- Let $r' = (1-b)/2c' > 0$, so $b + c'r' < 1$. Since p is a continuous concave function with $p(0) = 0$, $p(\frac{1}{2}) = \frac{1}{2}$, and since $p'(0) > 1$, we have $p(\eta) > \eta$ for all $\eta \in (0, \frac{1}{2})$; it follows that there is a finite constant k (depending only on f) such that $p^{(k)}(r) > \frac{1}{2} - r'$.

- Let $\delta' = 1/\ln(1/\delta)$. Whenever $\eta \leq r'$ we have $p(\frac{1}{2} - \eta) \geq \frac{1}{2} - (b + c'r')\eta$. Thus $p^{(k)}(\frac{1}{2} - r') \geq \frac{1}{2} - (b + c'r')^k r'$ for all k, and in particular, $p^{(k)}(\frac{1}{2} - r') \geq \frac{1}{2} - \delta'$ so long as k is at least
$$\frac{\ln(\delta'/r')}{\ln(b + c'r')} \leq O(\ln(1/\delta')) \leq o(\log_{1/b}\delta).$$

- Finally, by similar arguments we conclude that $p^{(k)}(\frac{1}{2} - \delta') \geq \frac{1}{2} - \delta$ so long as k is at least
$$\frac{\ln(\delta/\delta')}{\ln(b + c'\delta')} \leq \frac{\ln(\delta)}{(1 - O(\delta'))\ln b} \leq (1 + o(\delta))\log_{1/b}(1/\delta).$$

Combining the above five steps, we obtain the required result. □

Let us apply this result to the majority functions MAJ_k:

Theorem 3.7.2 *Let $k \geq 3$ be odd, and define*
$$a = k2^{1-k}\binom{k-1}{\frac{k-1}{2}} \geq (2/\pi)^{\frac{1}{2}}k^{\frac{1}{2}}, \qquad b = k2^{2-2k}\binom{k-1}{\frac{k-1}{2}}^2 \leq \frac{3}{4}.$$

Then $\mathsf{NS}_\epsilon(\mathsf{MAJ}_k^{\otimes \ell}) \geq \frac{1}{2} - \delta$ for $\ell \geq \Big(\log_a(1/\epsilon) + \log_{1/\delta}(1/\epsilon)\Big)(1 + r(\epsilon, \delta))$, where $r(\epsilon, \delta) \to 0$ as $\epsilon \to 0$ and $\delta \to 0$. Hence for every constant $\alpha < \frac{1}{2}$ and $\delta > 0$, there exists an odd $k \geq 3$ such that for ℓ sufficiently large, we have (writing $n = k^\ell$) the following:
$$\mathsf{NS}_{n^{-\alpha}}(\mathsf{MAJ}_k^{\otimes \ell}) \geq \frac{1}{2} - \delta.$$

Proof: This is an immediate corollary of Proposition 3.7.1, using the calculations from Proposition 3.4.6. □

In light of Corollary 3.6.4, we see that the iterated majority function is almost as sensitive to small noise as possible among monotone functions. We now give a slightly more involved construction that produces an explicit family of balanced monotone functions which have constant noise sensitivity at $\epsilon = n^{-1}\log^c n$, where $c < 1$. The functions are again defined by recursive majorities, but the arity is not constant; it increases with depth:

Definition 3.7.3 *The* expanding majority function*,* $\mathsf{EM}_n \colon \{+1,-1\}^n \to \{+1,-1\}$*, is defined for all* $n = 3^{2^\ell+\ell+1}$*,* $\ell \geq 1$*, by*

$$\mathsf{EM}_n = \mathsf{MAJ}_9 \otimes \mathsf{MAJ}_{27} \otimes \mathsf{MAJ}_{243} \otimes \cdots \otimes \mathsf{MAJ}_{3^{2^{\ell-1}+1}},$$

where the successive arities k_i *satisfy* $k_{i+1} = k_i^2/3$ *for all* $i = 1 \ldots \ell$*.*

We remark that the expanding majority function is balanced and also polynomial-time computable.

Theorem 3.7.4 *For every sufficiently small* $\epsilon > 0$*,*

$$\mathsf{NS}_{\epsilon/M}(\mathsf{EM}_n) \geq \epsilon - O(\epsilon^2),$$

where

$$M = n^{\frac{1}{2}}/\Theta(\log^t n), \qquad t = \log_2\sqrt{\pi/2} = .3257\ldots.$$

Proof: Recall that $\mathsf{EM}_n = \mathsf{MAJ}_{k_1} \otimes \cdots \otimes \mathsf{MAJ}_{k_\ell}$, where $k_1 = 3$ and $k_{i+1} = k_i^2/3$. We have n, the number of inputs, equal to $3^{2^\ell+\ell-1}$; hence $\log_2\log_3 n - 1 \leq \ell \leq \log_2\log_3 n$.

Let $\delta_0 = \epsilon/M$ (where M will be explicitly defined shortly), and recursively define $\delta_{i+1} = \mathsf{NS}_{\delta_i}(\mathsf{MAJ}_{k_{\ell-i}})$. Since all MAJ functions are balanced, Proposition 2.2.7 and the definition of EM_n tell us that $\mathsf{NS}_{\epsilon/M}(\mathsf{EM}_n) = \delta_\ell$. We will show that $\delta_\ell \geq \epsilon - O(\epsilon^2)$.

By Proposition 3.4.7,

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

where we define

$$g(t) = (2/\pi)^{\frac{1}{2}} t^{\frac{1}{2}} \exp(-1/3t).$$

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

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

Since noise sensitivity is an increasing function, 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

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

Now we define M:

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

which is $n^{\frac{1}{2}}/\Theta(\log^t n)$ as claimed. Recalling $\delta_0 = \epsilon/M$, 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 $\sum_{m=1}^{\ell}[k_m/g(k_1)g(k_2)\cdots g(k_m)] = O(1)$, we are 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 $k_{m-1} = (3k_m)^{\frac{1}{2}}$ by definition, so this ratio is $(k_m/3)^{\frac{1}{2}}/g(k_m) = (\pi/6)^{\frac{1}{2}}/\exp(-1/3k_m) < 1$. Hence the terms in the sum decrease geometrically, so the sum is indeed $O(1)$. □

3.8 Talagrand's random CNF

In [Tal96], Talagrand gives a probabilistic construction of a monotone boolean function $C_n\colon \{+1,-1\}^n \to \{+1,-1\}$ with the following property: at least an $\Omega(1)$ fraction of points x in $\{+1,-1\}^n$ satisfy both $C_n(x) = \mathrm{T}$ and $\#\{x'\colon \Delta(x,x') = 1 \text{ and } f(x) = \mathrm{F}\} \geq \Omega(n^{\frac{1}{2}})$, where Δ denotes Hamming distance. It is natural to conjecture that this function is sensitive to noise as small as $n^{-\frac{1}{2}}$, and indeed we shall now prove this.

Talagrand's function $C = C_n$ is a random CNF formula on its n inputs. Specifically, C is the $2^{n^{\frac{1}{2}}}$-wise AND of $n^{\frac{1}{2}}$-wise ORs, where each OR's inputs are selected independently and uniformly at random (with replacement) from $[n]$.

Theorem 3.8.1 *Talagrand's construction shows there exists an infinite family of monotone functions* (C_n) *satisfying*

$$\mathsf{NS}_{n^{-\frac{1}{2}}}(C_n) \geq \Omega(1).$$

Proof: Write $\epsilon = n^{-\frac{1}{2}}$. To prove the theorem, we show that if we pick C, x, and $y = N_\epsilon(x)$ at random, then

$$\mathbf{E}_C\left[\mathbf{P}[C(x) \neq C(y)]\right] \geq \Omega(1).$$

We shall imagine first picking and fixing the two inputs x and y, and then choosing C at random. Let n_{FF} denote the number of indices i such that $x_i = \mathrm{F}$ and $y_i = \mathrm{F}$, let n_{FT} denote the number of indices i such that $x_i = \mathrm{F}$ and $y_i = \mathrm{T}$; analogously define n_{TF} and n_{TT}. Also define $n_{\mathrm{F}\star}$ to be $n_{\mathrm{FF}} + n_{\mathrm{FT}}$, the number of F's in x, and similarly $n_{\star\mathrm{F}}, n_{\mathrm{T}\star}, n_{\star\mathrm{T}}$.

Now consider the construction of C; in particular, look at some particular OR, call it $\vee$. Let p_{FF} denote the probability — over the choice of C — that $\vee(x) = \mathrm{F}$ and $\vee(y) = \mathrm{F}$. Again, define $p_{\mathrm{FT}}, p_{\mathrm{TF}}, p_{\mathrm{TT}}, p_{\mathrm{F}\star}$, etc. analogously.

We immediately get

$$\begin{aligned} p_{\mathrm{FF}} &= \left(\frac{n_{\mathrm{FF}}}{n}\right)^{n^{\frac{1}{2}}}, \\ p_{\mathrm{TF}} &= \left(\frac{n_{\star\mathrm{F}}}{n}\right)^{n^{\frac{1}{2}}} - \left(\frac{n_{\mathrm{FF}}}{n}\right)^{n^{\frac{1}{2}}}, \\ p_{\mathrm{FT}} &= \left(\frac{n_{\mathrm{F}\star}}{n}\right)^{n^{\frac{1}{2}}} - \left(\frac{n_{\mathrm{FF}}}{n}\right)^{n^{\frac{1}{2}}}, \end{aligned} \tag{3.12}$$

and by subtracting these quantities from 1,

$$p_{\mathrm{TT}} \quad = \quad 1 - \left(\frac{n_{\star\mathrm{F}}}{n}\right)^{n^{\frac{1}{2}}} - \left(\frac{n_{\mathrm{F}\star}}{n}\right)^{n^{\frac{1}{2}}} + \left(\frac{n_{\mathrm{FF}}}{n}\right)^{n^{\frac{1}{2}}}.$$

All of the ORs are independent, so we may make a similar calculation for the main AND in f; call it $\wedge$. Let q_{FF} denote the probability — over the choice of C still — that $\wedge(x) = \mathrm{F}, \wedge(y) = \mathrm{F}$, and again define q_{FT}, etc.

Calculating as before,

$$\begin{aligned} q_{\mathrm{TT}} &= p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}, \\ q_{\mathrm{TF}} &= p_{\mathrm{T}\star}^{2^{n^{\frac{1}{2}}}} - p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}, \\ q_{\mathrm{FT}} &= p_{\star 1}^{2^{n^{\frac{1}{2}}}} - p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}. \end{aligned}$$

Now the probability that $C(x) \neq C(y)$ is simply $q_{\mathrm{FT}} + q_{\mathrm{TF}}$. Hence

$$\begin{aligned} \mathbf{E}_C\big[\mathbf{P}[C(x) \neq C(y)]\big] &= \mathbf{E}_{x,\, y=N_\epsilon(x)}\big[\mathbf{P}_C[C(x) \neq C(y)]\big] \\ &= \mathbf{E}_{x,y}[q_{\mathrm{FT}} + q_{\mathrm{TF}}] \\ &= \mathbf{E}_{x,y}[q_{\mathrm{FT}}] + \mathbf{E}_{x,y}[q_{\mathrm{TF}}]. \end{aligned} \tag{3.13}$$

Since x and y have the same distribution, $\mathbf{E}_{x,y}[q_{\mathrm{FT}}] = \mathbf{E}_{x,y}[q_{\mathrm{TF}}]$ by symmetry. Hence (3.13) $= 2\mathbf{E}_{x,y}[q_{\mathrm{TF}}]$. Thus it suffices to show

$$\mathbf{E}_{x,y}[q_{\mathrm{TF}}] = \mathbf{E}_{x,y}[p_{\mathrm{T}\star}^{2^{n^{\frac{1}{2}}}} - p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}] \geq \Omega(1).$$

We now focus on the quantity $(*) = p_{\mathrm{T}\star}^{2^{n^{\frac{1}{2}}}} - p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}$. Let $g(t) = t^{2^{n^{\frac{1}{2}}}}$. By the mean value theorem, $g(b) - g(a) = (b-a)g'(c)$ for some $c \in [a,b]$. Thus

$$(*) = (p_{\mathrm{T}\star} - p_{\mathrm{TT}})2^{n^{\frac{1}{2}}} c^{2^{n^{\frac{1}{2}}}-1} = 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} c^{2^{n^{\frac{1}{2}}}-1}$$

for some $c \in [p_{\mathrm{TT}}, p_{\mathrm{T}\star}]$.

Since $c^{2^{n^{\frac{1}{2}}}-1}$ is no smaller than $p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}$, we conclude

$$(*) \geq 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} p_{\mathrm{TT}}^{2^{n^{\frac{1}{2}}}}. \tag{3.14}$$

We proceed by conditioning on $(n_{\mathrm{FF}}, n_{\mathrm{FT}}, n_{\mathrm{TF}}, n_{\mathrm{TT}})$. Since $n_{\mathrm{F}\star} \sim \mathrm{Bin}(n, \frac{1}{2})$, the probability that $n_{\mathrm{F}\star}$ is outside the range $[n/2 - n^{\frac{1}{2}}, n/2 + n^{\frac{1}{2}}]$ is at most .05, for sufficiently large n (by a standard tail bound; $n^{\frac{1}{2}}$ is two standard deviations). Assuming that $n_{\mathrm{F}\star}$ is some fixed quantity in this range, $n_{\mathrm{FF}} \sim \mathrm{Bin}(n_{\mathrm{F}\star}, 1 - \epsilon)$. By a similar tail bound, the probability that n_{FF} is larger than $(1 - \epsilon + 2(\epsilon/n_{\mathrm{F}\star})^{\frac{1}{2}})n_{\mathrm{F}\star}$ is again at most .05. So assuming n is sufficiently large, we have that except with probability .1,

$$\begin{aligned} n_{\mathrm{F}\star} &\in \left[n/2 - n^{\frac{1}{2}}, n/2 + n^{\frac{1}{2}}\right], & (3.15) \\ n_{\mathrm{FF}}/n_{\mathrm{F}\star} &< 1 - \epsilon + 3(\epsilon/n)^{\frac{1}{2}}, & (3.16) \end{aligned}$$

where (3.16) uses the bound $n_{\mathrm{F}\star} > n/2.1$.

Finally, just as $n_{\mathrm{F}\star} \in [n/2 - n^{\frac{1}{2}}, n/2 + n^{\frac{1}{2}}]$ except with probability .05, so too may we conclude

$$n_{\star\mathrm{F}} \in \left[\frac{n}{2} - n^{\frac{1}{2}}, \frac{n}{2} + n^{\frac{1}{2}}\right], \tag{3.17}$$

except with probability .05.

In conclusion, (3.15), (3.16), and (3.17) hold, except with probability at most .15. Since $(*) \geq 0$ always,

$$\begin{aligned} \mathbf{E}_{x,y}[(*)] &\geq \mathbf{E}_{x,y}[(*) \mid (3.15), (3.16), (3.17)] \times \mathbf{P}[(3.15), (3.16), (3.17)] \\ &\geq .85\, \mathbf{E}_{x,y}[(*) \mid (3.15), (3.16), (3.17)]. \end{aligned}$$

Since we are only trying to prove $\mathbf{E}_{x,y}[(*)] \geq \Omega(1)$, we will henceforth assume (3.15), (3.16), and (3.17) hold, and it suffices to prove $\mathbf{E}_{x,y}[(*)] \geq \Omega(1)$ conditioned on this assumption. That is, all future expectations are conditioned on (3.15), (3.16), and (3.17).

Continuing from (3.14),

$$
\begin{aligned}
(*) \quad &\geq 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} \left(1 - \left(\tfrac{n_{\star \mathrm{F}}}{n}\right)^{n^{\frac{1}{2}}} - \left(\tfrac{n_{\mathrm{F}\star}}{n}\right)^{n^{\frac{1}{2}}} + \left(\tfrac{n_{\mathrm{FF}}}{n}\right)^{n^{\frac{1}{2}}}\right)^{2^{n^{\frac{1}{2}}}} \\
&\geq 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} \left(1 - \left(\tfrac{n_{\star \mathrm{F}}}{n}\right)^{n^{\frac{1}{2}}} - \left(\tfrac{n_{\mathrm{F}\star}}{n}\right)^{n^{\frac{1}{2}}}\right)^{2^{n^{\frac{1}{2}}}} \\
&\geq 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} \left(1 - (\tfrac{1}{2} + n^{-\frac{1}{2}})^{n^{\frac{1}{2}}} - (\tfrac{1}{2} + n^{-\frac{1}{2}})^{n^{\frac{1}{2}}}\right)^{2^{n^{\frac{1}{2}}}} && \text{(by (3.15) and (3.17))} \\
&\geq 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} (1 - 2e/2^{n^{\frac{1}{2}}})^{2^{n^{\frac{1}{2}}}} && \text{(asymptotically)} \\
&\geq e^{-2e} 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} \\
&\geq .004 \cdot 2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} && \text{(for all } n\text{)}
\end{aligned}
$$

Hence showing $\mathbf{E}_{x,x'}[(*)] \geq \Omega(1)$ amounts to showing $\mathbf{E}_{x,x'}[2^{n^{\frac{1}{2}}} p_{\mathrm{TF}}] \geq \Omega(1)$. By (3.12),

$$
\begin{aligned}
2^{n^{\frac{1}{2}}} p_{\mathrm{TF}} &= \Big(2\frac{n_{\star \mathrm{F}}}{n}\Big)^{n^{\frac{1}{2}}} \Big(1 - \Big(\frac{n_{\mathrm{FF}}}{n_{\star 0}}\Big)^{n^{\frac{1}{2}}}\Big) \\
&\geq (1 - 2n^{-\frac{1}{2}})^{n^{\frac{1}{2}}} \Big(1 - \Big(\frac{n_{\mathrm{FF}}}{n_{\star \mathrm{F}}}\Big)^{n^{\frac{1}{2}}}\Big) && \text{(by (3.17))} \\
&\geq e^{-2} \Big(1 - \Big(\frac{n_{\mathrm{FF}}}{n_{\star \mathrm{F}}}\Big)^{n^{\frac{1}{2}}}\Big) \\
&\geq e^{-2} \Big(1 - (1 - \epsilon + 2(\epsilon/n)^{\frac{1}{2}})^{n^{\frac{1}{2}}}\Big) && \text{(by (3.16))}
\end{aligned}
$$

Recalling $\epsilon = n^{-\frac{1}{2}}$, the quantity $(1 - \epsilon + 2(\epsilon/n)^{\frac{1}{2}})^{n^{\frac{1}{2}}}$ is asymptotically e^{-1} for large n. Hence

$$
\mathbf{E}_{x,y}[2^{n^{\frac{1}{2}}} p_{\mathrm{TF}}] \quad \geq e^{-2}(1 - e^{-1}) \geq \Omega(1),
$$

as claimed. □

We conclude that Corollary 3.6.4 is tight up to a a constant.

3.9 Tribes functions

In this section we consider read-once CNF; that is, functions of the form $f = \mathsf{AND}_a \otimes \mathsf{OR}_b$. By combining Propositions 3.3.1 and 3.3.2, we get

Proposition 3.9.1 *Let $f = \mathsf{AND}_a \otimes \mathsf{OR}_b$ and write $n = ab$. Then*

$$\mathsf{NS}_\epsilon(f) = 2\mathbf{P}[f = \mathrm{T}] \left(1 - \left(1 - \frac{1-(1-\epsilon)^b}{2^b(1-2^{-b})}\right)^a\right).$$

Hence

$$\mathsf{NS}'_\epsilon(f) = \frac{2\mathbf{P}[f = \mathrm{T}]n}{2^b - 1}(1-\epsilon)^{b-1}\left(1 - \frac{1-(1-\epsilon)^b}{2^b-1}\right)^{a-1},$$

and in particular,

$$\mathsf{NS}'_0(f) = \frac{2\mathbf{P}[f = \mathrm{T}]n}{2^b-1} \qquad \mathsf{NS}'_{\frac{1}{2}}(f) = \frac{(2\mathbf{P}[f = \mathrm{T}])^2 n}{(2^b-1)^2}.$$

The quantities $\mathsf{NS}'_0(f)$ and $\mathsf{NS}'_{\frac{1}{2}}(f)$ can also be calculated easily using Proposition 2.3.7 and Mansour's calculation of the Fourier coefficients $\mathsf{AND}_a \otimes \mathsf{OR}_b$ from [Man95]:

Proposition 3.9.2 *Let $f = \mathsf{AND}_a \otimes \mathsf{OR}_b$. Then*

$$\hat{f}(\emptyset) = 1 - 2\mathbf{P}[f = \mathrm{T}], \text{ and}$$

$$\hat{f}(S_1, \ldots, S_a) = 2(-1)^{c+1}2^{-bc}(1-2^{-b})^{a-c},$$

where $S = (S_1, \ldots, S_a)$ denotes the natural partition of a set $S \subseteq [n]$ according to the input blocks, and $c \geq 1$ is the number of $S_i \neq \emptyset$.

It is natural to select a and b in such a way that f has $\mathbf{P}[f = T] = (1-2^{-b})^a$ as close to $\frac{1}{2}$ as possible. To do this, it is best to choose a particular value for $n = ab$ for each possible value of b. Doing this, we get an infinite family of functions called the "tribes functions." These functions were first defined and studied by Ben-Or and Linial [BL90], who showed these functions satisfy $I_j(f) = \frac{\ln n}{n}(1 - o(1))$ for every $j \in [n]$.

Definition 3.9.3 *Given $b \in \mathbf{N}$, define $n = n_b$ to be the smallest integral multiple of b such that $(1-2^{-b})^{n/b} \leq \frac{1}{2}$. We then define the* tribes *function on n inputs to be $\mathsf{TRIBES}_n = \mathsf{AND}_{n/b} \otimes \mathsf{OR}_b$.*

We emphasize that the tribes functions are only defined for some input lengths n.

Let us quantify the relationship between b and n more carefully. For analysis purposes, let n' be the real number such that $(1-2^{-b})^{n'/b} = \frac{1}{2}$. Then $b = \log_2 n' - \log_2 \ln n' + o(1)$; hence $n' \le n \le n' + \log_2 n'$, and so $b = \log_2 n - \log_2 \ln n + o(1)$ as well. Thus $\mathbf{P}[\mathsf{TRIBES}_n = \mathrm{T}]$ is equal to $(1-2^{-b})^{n/b} = \frac{1}{2}(1-2^{-b})^{(n-n')/b} = \frac{1}{2}(1-2^{-b})^{1+o(1)} = \frac{1}{2}(1-O(\frac{\log n}{n}))$. Summarizing:

Fact 3.9.4 *Fix b and let n be chosen as in the definition of TRIBES_n. Then*

- $\mathbf{P}[\mathsf{TRIBES}_n = \mathrm{T}] = \frac{1}{2} - O(\frac{\log n}{n})$,
- $n = (1 + o(1))(\ln 2)b2^b$ *and hence* $n_{b+1} = (2 + o(1))n_b$,
- $b = \log_2 n - \log_2 \ln n + o(1)$, *so* $2^b = \frac{n}{\ln n}(1 + o(1))$.

The tribes functions are very noise-sensitive monotone functions, especially for ϵ near $\frac{1}{2}$. Combining Proposition 3.9.1, Fact 3.9.4, and Proposition 2.3.7 we get the following:

Proposition 3.9.5

$$\begin{aligned}\mathsf{NS}'_0(\mathsf{TRIBES}_n) = I(\mathsf{TRIBES}_n) &= (\ln n)(1 - o(1)),\\ \mathsf{NS}'_{\frac{1}{2}}(\mathsf{TRIBES}_n) = II(\mathsf{TRIBES}_n) &= \frac{\ln^2 n}{n}(1 - o(1)).\end{aligned}$$

These facts were known to Ben-Or and Linial. As Kahn et al. point out, the tribes function are tight (up to a constant) for Theorem 2.4.4; i.e., they have minimal $\mathsf{NS}'_{\frac{1}{2}} = II$ among near-balanced monotone functions. Some noise sensitivity graphs of tribes functions are shown in Figure 3-9.

As mentioned in Section 3.6, not only is $\mathsf{NS}'_{\frac{1}{2}}(\mathsf{TRIBES})$ very small, but $\mathsf{NS}_{\frac{1}{2}-\delta}$ is nearly as large as $\frac{1}{2} - \frac{\ln^2 n}{n}\delta$ for a reasonably large range of δ's:

Proposition 3.9.6

$$\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \ge \frac{1}{2} - \frac{\ln^2 n}{n}\,\delta \cdot (2 + o(1))(1 + 2\delta)^{\log_2 n} - O\left(\frac{\log^2 n}{n^2}\right).$$

Proof: By Facts 2.1.8 and 3.9.4, $\mathsf{NS}_{\frac{1}{2}}(\mathsf{TRIBES}_n) = \frac{1}{2} - 2\mathbf{P}[\mathsf{TRIBES}_n = \mathrm{T}]^2$, which is $\frac{1}{2} - O(\frac{\log^2 n}{n^2})$. Since $\mathsf{NS}_\epsilon(\mathsf{TRIBES}_n)$ is an increasing concave function of ϵ (Proposition 2.1.9),

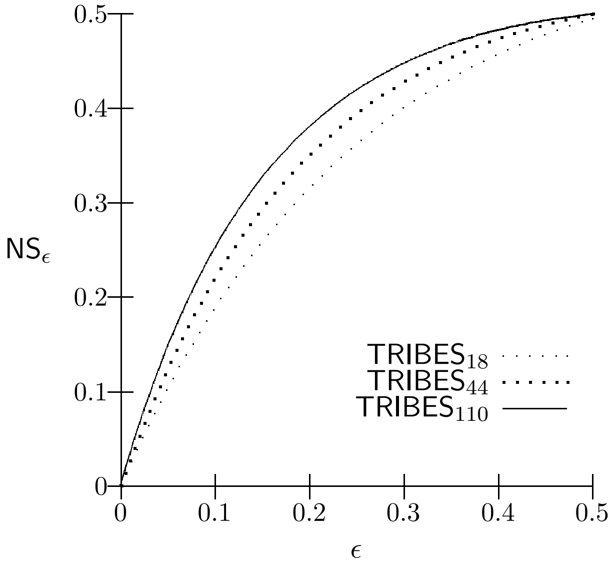

Figure 3-9: Noise sensitivity graphs of TRIBES

we get $\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \geq \frac{1}{2} - O(\frac{\log^2 n}{n}) - \delta \mathsf{NS}'_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n)$ by the mean value theorem. It remains to check that $\mathsf{NS}'_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \leq \frac{\ln^2 n}{n} \cdot (2+o(1))(1+2\delta)^{\log_2 n}$. By Proposition 3.9.1,

$$\begin{aligned}
\mathsf{NS}'_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) &= \frac{2\mathbf{P}[f=\mathrm{T}]n}{2^b-1}\left(\frac{1}{2}+\delta\right)^{b-1}\left(1-\frac{1-(\frac{1}{2}+\delta)^b}{2^b-1}\right)^{n/b-1} \\
&\leq \frac{2\mathbf{P}[f=\mathrm{T}]n}{2^b-1}\left(\frac{1}{2}+\delta\right)^{b-1} \\
&\leq \frac{2n}{2^b-1}\left(\frac{1}{2}\right)^{b}(1+2\delta)^{b-1} \\
&\leq \frac{\ln^2 n}{n}(2+o(1))(1+2\delta)^{\log_2 n},
\end{aligned}$$

where we have used Fact 3.9.4. □

We remark that, as might be expected, a slightly more careful analysis lets one shave off a factor of 2; one can show

$$\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \geq \frac{1}{2} - \frac{\ln^2 n}{n}\,\delta \cdot (1+o(1))(\eta + O(\eta^2)) - O\left(\frac{\log^2 n}{n^2}\right),$$

where $\eta = (1+2\delta)^{\log_2 n}$. However, we shall not bother to prove this. An alternate way of deriving these estimates based on the Fourier spectrum of the tribes function is given in [O'D02].

As an immediate corollary of Proposition 3.9.6, we have the following:

Corollary 3.9.7 *If $\delta \leq O(\frac{1}{\log n})$ then $\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{TRIBES}_n) \geq \frac{1}{2} - O(\frac{\log^2 n}{n}\ \delta)$.*

Thus in the range $\delta \leq O(\frac{1}{\log n})$, Corollary 3.6.5 is tight up to a constant. If we are concerned about the range $\delta \leq \frac{1}{2} - \Omega(1)$, we can start by iterating MAJ_3 and then use tribes:

Theorem 3.9.8 *Let $\epsilon \geq \Omega(1)$. Then there is an infinite family of monotone functions (T_n) satisfying*

$$\mathsf{NS}_\epsilon(T_n) \geq \frac{1}{2} - \frac{\log^{1+u'} n}{n},$$

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

Proof: As stated, the idea is to first use iterated MAJ_3 to boost ϵ up to $\frac{1}{2} - \frac{1}{\log n}$, and then to apply a tribes function.

Fix a tribes function on n inputs, TRIBES_n. We will construct $T_{n'}$ on $n' = n \log^{u'} n$ inputs. Let ℓ be the recursive depth necessary for iterated MAJ_3's to increase noise ϵ up to noise $\frac{1}{2} - \frac{1}{\log n}$. By Theorem 3.7.2, $\ell = (1 + o(1)) \log_{\frac{4}{3}}(\log n)$ is sufficient, since $\epsilon \geq \Omega(1)$. Put $f = \mathsf{MAJ}_r^{\otimes \ell}$, so f is a function on $3^\ell = \log^{u'} n$ inputs. Let $T_{n'} = \mathsf{TRIBES}_n \otimes f$.

Now by construction we have $\mathsf{NS}_\epsilon(f) \geq \frac{1}{2} - \frac{1}{\log n}$. By Corollary 3.9.7 we have, $\mathsf{NS}_{\frac{1}{2}-1/\log n}(\mathsf{TRIBES}_n) \geq O(\frac{\log n}{n})$. Since f is balanced, by Proposition 2.2.7 we get $\mathsf{NS}_\epsilon(T_{n'}) \geq \frac{1}{2} - O(\frac{\log n}{n})$. The result now follows, since as a function of n', $O(\frac{\log n}{n})$ is in fact $\frac{\log^{1+u'} n'}{n'}$ (taking u' slightly larger to kill any constant factors). □

It seems possible to improve Theorem 3.9.8 to $\frac{1}{2} - \frac{\log^{2+o(1)} n}{n}$ by using recursive tribes functions in the manner the expanding majority functions. However the fact that the tribes functions are not perfectly balanced is a stumbling block.

Let us conclude this section with a brief estimate of the ℓ_2 weight of the Fourier spectrum of the tribes functions at low levels. From Proposition 3.9.2 we know that an index set $(S_1, \ldots, S_{n/b})$ with $c \geq 1$ nonempty parts contributes

$$\begin{aligned} 4 \cdot 2^{-2bc}(1 - 2^{-b})^{2(n/b-c)} &= 4\mathbf{P}[\mathsf{TRIBES}_n = \mathrm{T}]^2 \left(\frac{2^{-b}}{1 - 2^{-b}}\right)^{2c} \\ &= \left(1 - O\left(\frac{\log n}{n}\right)\right) \left(\frac{\ln n}{n}\right)^{2c} (1 + o(1))^{2c} \end{aligned}$$

in ℓ_2 weight to the Fourier spectrum. For a given level $k \geq 1$, let us lower-bound $\sum_{|S|=k} \widehat{\mathsf{TRIBES}}_n(S)^2$ simply by taking the contribution of indices with $c = 1$. There are exactly $\frac{n}{b}\binom{b}{k}$ such indices, and each contributes weight (at least) $(1 - o(1))\frac{\ln^2 n}{n^2}$. For $k = o(b^{\frac{1}{2}}) = \log^{\frac{1}{2}} n$ we have $\binom{b}{k} \geq (1 - o(1))\frac{b^k}{k!}$; hence we have the following conclusion:

Proposition 3.9.9 *For* $1 \leq k < o(\log^{\frac{1}{2}} n)$*,*

$$\sum_{|S|=k} \widehat{\mathsf{TRIBES}}_n(S)^2 \geq (1 - o(1))\frac{\ln^2 n}{n}(\log_2 n)^{k-1}.$$

This shows that the estimate of Benjamini, Kalai, and Schramm, Theorem 2.4.8, has the correct asymptotics, since in the case of the tribes functions $I\!I \sim \frac{\ln^2 n}{n}$.

A representative Fourier spectrum graph of the tribes function is shown in Figure 3-10.

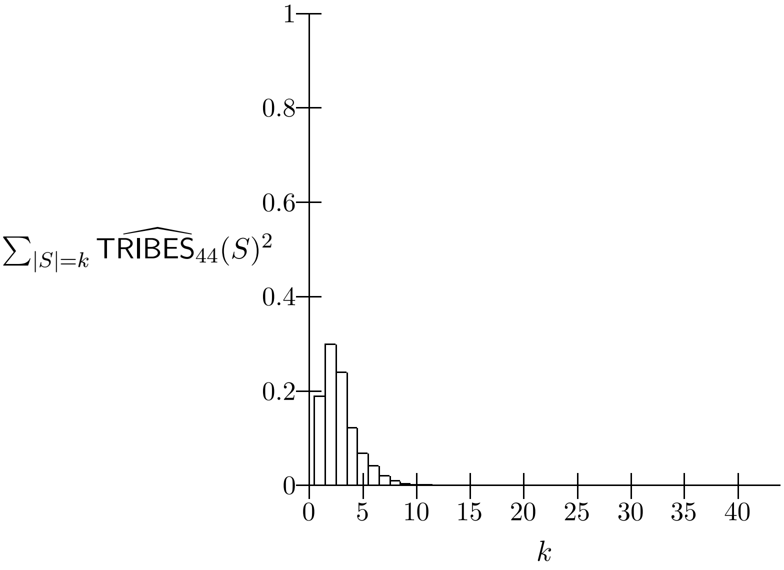

Figure 3-10: Fourier spectrum graph of TRIBES_{44}

3.10 Threshold functions

In this section we investigate the sensitivity of weighted threshold functions to small rates of noise. Benjamini, Kalai, and Schramm gave a somewhat difficult probabilistic argument in [BKS99] showing that the noise sensitivity at ϵ of every threshold function is bounded above by $O(\epsilon^{\frac{1}{4}})$, where the hidden constant is *independent of n*. Peres gave a much simpler argument improving this to the expected tight upper bound $O(\sqrt{\epsilon})$ (unpublished; see [BKS99]). We now present a slightly altered version of Peres's proof from [Per02]:

Lemma 3.10.1 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a weighted threshold function, say $f(x) = \mathrm{sgn}(\sum_{i=1}^n w_i x_i - \theta)$, and assume without loss of generality that $\sum_{i\in F} w_i x_i = 0$ never holds for any x and $\emptyset \neq F \subseteq [n]$ (perturb the weights if necessary). Then for any set $\emptyset \neq F \subseteq [n]$,*

$$\mathop{\mathbf{P}}_{x,y=N_\epsilon(x)}[f(x) \neq f(y) \mid \textit{the flipped bits are precisely } F] = \mathbf{E}\left[f(x)\mathrm{sgn}\left(\sum_{i\in F} w_i x_i\right)\right].$$

Proof: Fix $\emptyset \neq F \subseteq [n]$. Let x be chosen uniformly at random, and let y be given by flipping the F bits of x. Write $S = \sum_{i=1}^n w_i x_i$ and $S' = \sum_{i=1}^n w_i y_i$, and note that S and S' are identically distributed, although they are not independent.

Suppose we condition on $\mathrm{sgn}(S-\theta) = \mathrm{sgn}(S'-\theta)$. Then S and S' are still identically distributed; hence we can imagine choosing S and S' by first choosing the unordered pair $\{S, S'\}$ subject to $\mathrm{sgn}(S-\theta) = \mathrm{sgn}(S'-\theta)$, and then flipping a fair coin to decide which is which. The value of $\mathrm{sgn}(S-\theta)$ is the same regardless of the coin toss, and thus $\mathrm{sgn}(S-S')$ is equally likely ±1, independent of $\mathrm{sgn}(S-\theta)$. We have thus established the following:

$$\mathbf{E}[\mathrm{sgn}(S-\theta)\mathrm{sgn}(S-S') \mid \mathrm{sgn}(S-\theta) = \mathrm{sgn}(S'-\theta)] = 0. \tag{3.18}$$

Suppose on the other hand we condition on $\mathrm{sgn}(S-\theta) = \mathrm{sgn}(S'-\theta)$. Then we know that θ is between the two numbers S and S', and so $\mathrm{sgn}(S-\theta) = \mathrm{sgn}(S-S')$. Hence

$$\mathbf{E}[\mathrm{sgn}(S-\theta)\mathrm{sgn}(S-S') \mid \mathrm{sgn}(S-\theta) \neq \mathrm{sgn}(S'-\theta)] = 1. \tag{3.19}$$

Combining Equations (3.18) and (3.19), we get

$$\begin{array}{rrcl} & \mathbf{P}[\mathrm{sgn}(S-\theta) \neq \mathrm{sgn}(S'-\theta)] & = & \mathbf{E}[\mathrm{sgn}(S-\theta)\mathrm{sgn}(S-S')] \\ \Rightarrow & \mathbf{P}[f(x) \neq f(y)] & = & \mathbf{E}\left[f(x)\mathrm{sgn}\left(2\sum_{i=1}^n w_i x_i\right)\right], \end{array}$$

and this is conditioned on F being the bits flipped in going from x to y, as desired. □

Theorem 3.10.2 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a weighted threshold function. Then*

$\mathsf{NS}_\epsilon(f) \leq (\sqrt{2/\pi} + o_\epsilon(1) + 1/n)\sqrt{\epsilon}$, *where* $o_\epsilon(1)$ *represents a function which goes to* 0 *as* $\epsilon \to 0$, *independent of* n.

Proof: Let f be as in Lemma 3.10.1, and for $F \neq \emptyset$, let $\rho_F = \mathbf{E}[f(x)\mathrm{sgn}(\sum_{i\in F} w_i x_i)]$. For each $0 < k \leq n$ we wish to upper-bound the probability $P(k)$ that $f(x) \neq f(N_\epsilon(x))$ given that the noise operator flips exactly k bits. By Lemma 3.10.1 we have $P(k) = \mathrm{avg}_{|F|=k}\{\rho_F\}$.

Write $m = \lfloor n/k \rfloor$, and let $\Pi = (F_1, \ldots, F_m)$ be a random partition of $[n]$ into m disjoint sets of size k, where $n - mk$ indices are left over. Since each F_j is uniformly distributed among all k-sets of $[n]$, we have

$$\begin{aligned} P(k) &= \operatorname*{avg}_{|F|=k} \rho_F \\ &= \frac{1}{m}\sum_{j=1}^{m} \mathbf{E}_\Pi[\rho_{F_j}] \\ &= \frac{1}{m}\mathbf{E}_\Pi\Big[\sum_{j=1}^{m} \rho_{F_j}\Big] \\ &= \frac{1}{m}\mathbf{E}_\Pi\Big[\sum_{j=1}^{m} \mathbf{E}_x\Big[f(x)\mathrm{sgn}\big(\sum_{i\in F_j} w_i x_i\big)\Big]\Big]. \end{aligned}$$

Let us write $\sigma_j(x) = \mathrm{sgn}(\sum_{i\in F_j} w_i x_i)$. Continuing:

$$\begin{aligned} P(k) &= \frac{1}{m}\mathbf{E}_\Pi\Big[\sum_{j=1}^{m} \mathbf{E}_x[f(x)\sigma_j(x)]\Big] \\ &= \frac{1}{m}\mathbf{E}_{\Pi,x}\Big[f(x)\Big(\sum_{j=1}^{m}\sigma_j(x)\Big)\Big]. \end{aligned} \tag{3.20}$$

Since $f(x) \in \{+1, -1\}$, regardless of the manner in which $f(x)$ is dependent on the random variables $\sigma_j(x)$, we have $f(x)(\sum_{j=1}^{m}\sigma_j(x)) \leq |\sum_{j=1}^{m}\sigma_j(x)|$. Hence

$$P(k) \leq \frac{1}{m}\mathbf{E}_{\Pi,x}\Big[\Big|\sum_{j=1}^{m}\sigma_j(x)\Big|\Big].$$

But since the sets F_j are always chosen pairwise disjoint, the random variables $\sigma_j(x)$ are mutually independent and uniformly ±1. It is well known that for independent random signs σ_j, $\mathbf{E}[|\sum_{j=1}^{m}\sigma_j|] \leq \sqrt{2/\pi}\,m^{\frac{1}{2}} + C$ for some small universal constant C. Thus

$$\begin{aligned} P(k) &\leq \sqrt{2/\pi}\,m^{-\frac{1}{2}} + C\,m^{-1} \\ &= \sqrt{2/\pi}\,\lfloor n/k \rfloor^{-\frac{1}{2}} + C\,\lfloor n/k \rfloor^{-1} \\ &\leq \sqrt{2/\pi}(k/n + 2(k/n)^2)^{\frac{1}{2}} + C\,(k/n + 2(k/n)^2), \end{aligned}$$

where we have used the fact that $\lfloor x^{-1} \rfloor^{-1} \leq x + 2x^2$ for all $x \in (0,1]$. Note that this statement is also true for $k = 0$. We conclude,

$$\begin{aligned} \mathsf{NS}_\epsilon(f) &= \mathop{\mathbf{E}}_{k \sim \mathrm{Bin}(n,\epsilon)}[P(k)] \\ &\leq \mathbf{E}[\sqrt{2/\pi}(k/n + 2(k/n)^2)^{\frac{1}{2}} + C(k/n) + 2C(k/n)^2] \\ &\leq \sqrt{2/\pi}\sqrt{\mathbf{E}[k/n + 2(k/n)^2]} + C\,\mathbf{E}[k/n] + 2C\,\mathbf{E}[(k/n)^2] \qquad \text{(Cauchy-Schwarz)} \\ &= \sqrt{2/\pi}\sqrt{\epsilon + 2\epsilon/n} + C\epsilon + 2C\epsilon/n \qquad (3.21) \\ &\leq (\sqrt{2/\pi} + o_\epsilon(1) + 1/n)\sqrt{\epsilon}, \end{aligned}$$

as claimed. □

Benjamini, Kalai, and Schramm conjectured that majority is the threshold function which is most sensitive to noise. This seems very plausible despite the small gap between $(2/\pi)^{\frac{1}{2}}$ and $2/\pi$ in Corollary 3.4.4 and Theorem 3.10.2.

Corollary 3.10.3 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a weighted threshold function. Then $\mathsf{NS}_\epsilon(f) \leq \frac{5}{4}\sqrt{\epsilon}$, and further $\mathsf{NS}_\epsilon(f) \leq \sqrt{\epsilon}$ if $\epsilon \leq \frac{1}{100}$.*

Proof: It is possible to check by exhaustion that the statements are true for $n \leq 4$; thus we may assume $n \geq 5$. We now perform the estimation at the end of the proof of Theorem 3.10.2 more explicitly. The constant C there may be taken to be .21. Starting from (3.21),

$$\begin{aligned} \mathsf{NS}_\epsilon(f) &\leq \sqrt{2/\pi}\sqrt{\epsilon + 2\epsilon/n} + C\epsilon + 2C\epsilon/n \\ &\leq \sqrt{2/\pi}\sqrt{\epsilon + 2\epsilon/5} + .21\epsilon + .084\epsilon \\ &\leq \sqrt{2/\pi}\sqrt{7/5}\sqrt{\epsilon} + .294\epsilon. \qquad (3.22) \end{aligned}$$

It is straightforward to check that (3.22) is never more than $\frac{5}{4}\sqrt{\epsilon}$, and is at most $\sqrt{\epsilon}$ for $\epsilon \leq \frac{1}{100}$. □

While Peres's bound is sharp (up to a constant) for threshold functions in general, more can be said if we take into account the bias of the function.

Theorem 3.10.4 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a weighted threshold function, and let $p = \min\{\mathbf{P}[f = \mathrm{T}], \mathbf{P}[f = \mathrm{F}]\}$. There is a universal constant C (e.g., 16 suffices), such that*

$$\mathsf{NS}_\epsilon(f) \leq C \cdot (2p)\sqrt{\ln(1/p)}\sqrt{\epsilon}.$$

Proof: Suppose without loss of generality that $p = \mathbf{P}[f = -1] \leq \frac{1}{2}$. Proceed as in the proof of Theorem 3.10.2 until Equation (3.20), recalling that the random variables σ_j are independent and uniformly ± 1. Let T be a random variable denoting the number of σ_j's which are $+1$, so $T \sim \mathrm{Bin}(m, \frac{1}{2})$. For each $t = 0 \ldots m$, define $p_t = \mathbf{P}[f(x) = -1 \mid T = t]$, where the probability is taken over the choice of Π and x. Since the event $f(x) = -1$ is negatively correlated with the events $\sigma_j = +1$, we conclude that

$$1 \geq p_0 \geq p_1 \geq p_2 \geq \cdots \geq p_m \geq 0. \tag{3.23}$$

Also, by definition,
$$p = \sum_{t=0}^{m} \mathbf{P}[T = t] p_t. \tag{3.24}$$

Continuing from Equation (3.20), we have

$$\begin{aligned}
P(k) &= \frac{1}{m}\mathbf{E}_{\Pi,x}\left[f(x)\left(\sum_{j=1}^{m}\sigma_j\right)\right] \\
&= \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t]\mathbf{E}\left[f(x)\left(\sum_{j=1}^{m}\sigma_j\right) \mid T=t\right] \\
&= \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t]\mathbf{E}[f(x)(2t-m) \mid T=t] \\
&= \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t](2t-m)\mathbf{E}[f(x) \mid T=t] \\
&= \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t](2t-m)(1-2p_t) \\
&= \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t](2t-m) + \frac{1}{m}\sum_{t=0}^{m}\mathbf{P}[T=t]2p_t(m-2t) \\
&= \frac{4}{m}\sum_{t=0}^{m}\mathbf{P}[T=t]\left(\frac{m}{2}-t\right)p_t, \qquad (3.25)
\end{aligned}$$

where we have used $\mathbf{E}_T[2T - m] = 0$.

We will obtain an upper bound for $P(k)$ by maximizing (3.25) subject to (3.23) and (3.24). This is a linear programming problem. Hence the maximum occurs at a vertex, which in this case means the maximum occurs when, for an appropriate $1 \leq b \leq \frac{m}{2}$, we have $p_0 = p_1 = \cdots = p_{b-1} = 1$, $p_{b+1} = p_{b+2} = \cdots = p_m = 0$, and p_b is such that (3.24) is tight. (We have $b \leq \frac{m}{2}$ since $p \leq \frac{1}{2}$.) Henceforth we let the p_t's take on these values which maximize (3.25) and we will reason about the value of (3.25).

Our goal is now to show the following:

$$(3.25) \leq (C/2) \cdot (2p)\sqrt{\ln(1/p)}m^{-\frac{1}{2}}. \tag{3.26}$$

An easy case occurs if $p \leq 2^{-m}$. In this case $p_t = 0$ for all $t > 0$ and hence $(3.25) = 2p$. But $p \leq 2^{-m}$ implies $(C/2)\sqrt{\ln(1/p)}m^{-\frac{1}{2}} \geq 1$ (assuming $C \geq 2$), so (3.26) is proved. Assume then that $p > 2^{-m}$. We claim that it suffices to show (3.26) in the case that $p_b = 1$; i.e., the case that $p = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$ for some $b > 0$. For suppose that (3.26) holds in this case; then given any $p^* > 2^{-m}$ and associated b and p_t^*'s, we may write $p^* = (1-p_b)p^- + p_b p^+$, where $p^- = \sum_{t=0}^{b-1} \binom{m}{t} 2^{-m}$ and $p^+ = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$. One can easily check that the value of (3.25) associated to p^* is $(1 - p_b)(3.25)^- + p_b(3.25)^+$, where $(3.25)^-$ denotes the value of (3.25) for p^-, and similarly for $(3.25)^+$. But by assumption (3.26) holds for p^- and p^+; since $p\sqrt{\ln(1/p)}$ is a convex function of p, we have that (3.26) holds for p^* as well.

So we may assume that $p = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$; i.e., $p = \mathbf{P}[T \leq b]$, where as before, $T \sim \mathrm{Bin}(m, 1/2)$. Now we can rewrite (3.25) as

$$\frac{4p}{m}\mathbf{E}\left[\frac{m}{2} - T \mid T \leq b\right].$$

Thus showing (3.26) amounts to showing

$$\mathbf{E}\left[\frac{m}{2} - T \mid T \leq b\right] \leq (C/4)\sqrt{\ln(1/\mathbf{P}[T \leq b])}\, m^{\frac{1}{2}}, \tag{3.27}$$

for every $1 \leq b \leq \frac{m}{2}$. We shall show this is indeed true for $C = 16$ in the technical Lemma 3.10.5 which follows. We thus have $P(k) \leq (C/2)(2p)\sqrt{\ln(1/p)}m^{-\frac{1}{2}}$. The remainder of the proof proceeds as in the proof of Theorem 3.10.2, noting that we can upper-bound $(1 + 1/n)$ by 2. □

We now prove the technical lemma needed in Theorem 3.10.4.

Lemma 3.10.5 *Let $m \geq 1$ be an integer and let $T \sim \mathrm{Bin}(n, \frac{1}{2})$. Let $1 \leq b \leq \frac{m}{2}$, and let $q = \mathbf{P}[T \leq b]$. Then*

$$\mathbf{E}\left[\frac{m}{2} - T \mid T \leq b\right] \leq 4\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}.$$

Proof: We will make use of the Chernoff bound $\mathbf{P}[T \leq \frac{m}{2} - \delta\frac{m}{2}] < \exp(-m\delta^2/4)$ which holds for $0 \leq \delta \leq 1$. This immediately yields

$$b \geq \frac{m}{2} - \sqrt{\ln(1/q)}\, m^{\frac{1}{2}}. \tag{3.28}$$

The Chernoff bound also tells us that $\mathbf{P}[T < \frac{m}{2} - 2\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}] < q^4$. Hence

$$\mathbf{P}\left[T < \frac{m}{2} - 2\sqrt{\ln(1/q)}\sqrt{m} \mid T \leq \frac{m}{2}\right] < 2q^4.$$

We write d for $2\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}$ and α for $2q^4$, so we have $\mathbf{P}[T < \frac{m}{2} - d \mid T \leq \frac{m}{2}] < \alpha$. It follows from the log-concavity of the binomial distribution[1] that

$$\mathbf{P}\left[T < u - d \mid T \leq u\right] < \alpha$$

holds for every $u \leq \frac{m}{2}$. In particular this holds for $u = b$, $u = b - d$, $u = b - 2d$, $u = b - 3d$, etc., whence

$$\begin{aligned}
\mathbf{E}\left[b - T \mid T \leq b\right] &\leq d + \alpha(2d) + \alpha^2(3d) + \alpha^3(4d) + \cdots \\
&< \frac{d}{(1-\alpha)^2} \\
&= \frac{2\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}}{(1-2q^4)^2} \\
&\leq \frac{2\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}}{(1-2(\frac{1}{2})^4)^2} \qquad (3.29)\\
&< 3\sqrt{\ln(1/q)}\, m^{\frac{1}{2}}. \qquad (3.30)
\end{aligned}$$

[1] In particular, log-concavity implies the "new-is-better-than-used" property of mathematical reliability, which is exactly what we use here. See, e.g., [An95].

So we conclude

$$\begin{aligned}\mathbf{E}\left[\frac{m}{2} - T \mid T \le b\right] &= \frac{m}{2} - b + \mathbf{E}\left[b - T \mid T \le b\right] \\ &< 4\sqrt{\ln(1/q)}\, m^{\frac{1}{2}},\end{aligned}$$

by (3.28) and (3.30). □

As an aside, we believe the lemma is true with the improved constant $1/\sqrt{2}$; in particular, we conjecture that

$$\max_{1 \le b \le \frac{m}{2}} \frac{\sqrt{\ln(1/\mathbf{P}[T \le b])}\, m^{\frac{1}{2}}}{\mathbf{E}\left[\frac{m}{2} - T \mid T \le b\right]} \tag{3.31}$$

increases as a function of m. If this is true than we can get the improved constant by noting that (3.31) $\to 1/\sqrt{2}$ as $m \to \infty$, by a straightforward normal approximation.

3.11 Read-once intersections of halfspaces

In this section we consider *read-once intersections of halfspaces*; i.e., functions of the form $f = \mathsf{AND}(h_1, \ldots, h_k)$, where each function h_i is a weighted threshold function.

Theorem 3.11.1 *Let $f = \mathsf{AND}(h_1, \ldots, h_k)$, where each function h_i is a weighted threshold function and $k \ge 2$. Let C be the constant from Theorem 3.10.4. Then*

$$\mathsf{NS}_\epsilon(f) \le 2C\,(\epsilon \ln k)^{\frac{1}{2}}.$$

Proof: By Proposition 2.2.8 and Corollary 3.4.4, $\mathsf{NS}_\epsilon(f) \le k \cdot \frac{5}{4}\, \epsilon^{\frac{1}{2}}$. For $k \le 6$ this is smaller than $2C\,(\epsilon \ln k)^{\frac{1}{2}}$, so we assume $k \ge 7$.

Write $p_i = \mathbf{P}[h_i = \mathrm{T}]$ and $\eta_i = \mathsf{NS}_\epsilon(h_i)$. By Proposition 3.3.2, we have the following:

$$\begin{aligned}\frac{1}{2}\mathsf{NS}_\epsilon(f) &= \left(\prod_{i=1}^{k} p_i\right)\left(1 - \prod_{i=1}^{k}\left(1 - \frac{\eta_i}{2p_i}\right)\right) \\ &\le \left(\prod_{i=1}^{k} p_i\right)\left(\sum_{i=1}^{k} \frac{\eta_i}{2p_i}\right),\end{aligned} \tag{3.32}$$

where we have used the fact that $\prod_{i=1}^{k}(1 - x_i) \ge 1 - \sum_{i=1}^{k} x_i$ for any $x_1, \ldots, x_k \in [0,1]$ (note that $\frac{\eta_i}{2p_i} \le 1$ by Fact 2.1.7).

We would now like to break up (3.32) into parts depending on the various values of the p_i's. For $1 \le s \le t \le k$, let us write

$$U_{s\ldots t} = \left(\prod_{i=s}^{t} p_i\right)\left(\sum_{i=s}^{t} \frac{\eta_i}{2p_i}\right).$$

Our goal is to show $U_{1\ldots k} \le C\ (\epsilon \ln k)^{\frac{1}{2}}$. We begin by claiming that it suffices to assume all of the p_i's are at least $\frac{2}{3}$. For suppose that, say, $p_1 < \frac{2}{3}$. Then

$$U_{1\ldots k} = p_1 U_{2\ldots k} + \left(\prod_{i=1}^{k} p_i\right)\left(\frac{\eta_1}{2p_1}\right) \le \frac{2}{3}U_{2\ldots k} + \frac{1}{2}\eta_1 \le \frac{2}{3}U_{2\ldots k} + \frac{5}{8}\ \epsilon^{\frac{1}{2}},$$

where the last step uses Corollary 3.4.4. Hence if we can show $U_{2\ldots k} \le C\ (\epsilon \ln k)^{\frac{1}{2}}$, then $U_{1\ldots k} \le \frac{2}{3}C\ (\epsilon \ln k)^{\frac{1}{2}} + \frac{5}{8}\ \epsilon^{\frac{1}{2}} \le C\ (\epsilon \ln k)^{\frac{1}{2}}$, since $\frac{1}{3}C\ (\ln k)^{\frac{1}{2}} \ge \frac{1}{3}C(\ln 7)^{\frac{1}{2}} \ge \frac{5}{8}$. We can repeat this argument for any $p_i < \frac{2}{3}$; thus in showing $U_{1\ldots k} \le C\ (\epsilon \ln k)^{\frac{1}{2}}$ it suffices to assume $p_i \ge \frac{2}{3}$ for all i.

Without loss of generality, we may reorder indices so that $p_1, \ldots, p_\ell < 1 - \frac{1}{10k}$ and $p_{\ell+1}, \ldots, p_k \ge 1 - \frac{1}{10k}$ for some $0 \le \ell \le k$. Using the fact that $p_i \le 1$ for all i, it is easy to see that $U_{1\ldots k} \le U_{1\ldots \ell} + U_{\ell+1\ldots k}$. We now upper-bound each of these terms individually.

$$\begin{aligned}
U_{1\ldots\ell} &= \left(\prod_{i=1}^{\ell} p_i\right)\left(\sum_{i=1}^{\ell} \frac{\eta_i}{2p_i}\right) \\
&\le \left(\prod_{i=1}^{\ell} p_i\right)\left(\sum_{i=1}^{\ell} C\ \epsilon^{\frac{1}{2}}\ \frac{1-p_i}{p_i}\left(\ln\frac{1}{1-p_i}\right)^{\frac{1}{2}}\right) && \text{(Theorem 3.10.4)} \\
&\le \tfrac{3}{2}C\ (\epsilon \ln 10k)^{\frac{1}{2}} \left(\textstyle\prod_{i=1}^{\ell} p_i\right)\left(\textstyle\sum_{i=1}^{\ell}(1-p_i)\right) && \text{(using } \tfrac{2}{3} \le p_i \le 1 - \tfrac{1}{10k}\text{)} \\
&\le \tfrac{3}{2}C\ (\epsilon \ln 10k)^{\frac{1}{2}}\ \mu^\ell \ell(1-\mu) && \text{(where } \mu = \sqrt[\ell]{p_1 \cdots p_\ell}\text{, by the AM-GM inequality)} \\
&\le \tfrac{3}{2}C\ (\epsilon \ln 10k)^{\frac{1}{2}}\ e^{-1} && \text{(by elementary calculus, maximizing over } \mu \in [0,1]\text{).}
\end{aligned}$$

$$\begin{aligned}
U_{\ell+1\ldots k} &\le \left(\prod_{i=\ell+1}^{k} p_i\right)\left(\sum_{i=\ell+1}^{k} C\ \epsilon^{\frac{1}{2}}\ \frac{1-p_i}{p_i}\left(\ln\frac{1}{1-p_i}\right)^{\frac{1}{2}}\right) && \text{(as before)} \\
&\le (k-\ell)C\ \epsilon^{\frac{1}{2}}\ \frac{\frac{1}{10k}}{1-\frac{1}{10k}}(\ln 10k)^{\frac{1}{2}} && \text{(since } \tfrac{1-p}{p}(\ln\tfrac{1}{1-p})^{\frac{1}{2}} \text{ decreases on } [1-\tfrac{1}{10k}, 1]\text{)} \\
&\le (10 - \tfrac{1}{7})^{-1} C\ (\epsilon \ln 10k)^{\frac{1}{2}} && \text{(using } k-\ell \le k,\ k \ge 7\text{)}
\end{aligned}$$

Thus $U_{1\ldots k} \leq (\frac{3}{2e} + (10 - \frac{1}{7})^{-1})C\ (\epsilon \ln 10k)^{\frac{1}{2}}$ and this is less than $C\ (\epsilon \ln k)^{\frac{1}{2}}$ for $k \geq 7$, as desired. □

We remark that Theorem 3.11.1 is trivial unless ϵ is significantly smaller than $\frac{1}{\ln k}$.

3.12 Read-once majorities of halfspaces

In this section we consider *read-once unweighted thresholds of halfspaces*; i.e., functions of the form $f = \mathrm{sgn}(h_1 + \cdots h_k - \theta)$, where the functions h_i are weighted threshold function on disjoint sets of variables. We mainly have in mind functions of the form $\mathsf{MAJ}_k(h_1, \ldots, h_k)$.

This section shall be devoted to the proof of the following theorem:

Theorem 3.12.1 *Let $f(x) = \mathrm{sgn}(h_1(x) + \cdots + h_k(x) - \theta)$, where the functions h_i are weighted threshold functions on disjoint sets of variables. Then*

$$\mathsf{NS}_\epsilon(f) \leq \tilde{O}\left((\epsilon \log \tfrac{k}{\epsilon})^{\frac{1}{4}}\right).$$

(Here $\tilde{O}(\eta)$ denotes $O(1) \cdot \eta \log(1/\eta)$.)

Proof: Let $p_i = \mathbf{P}[h_i(x) = -1]$ and $q_i = \min\{p_i, 1 - p_i\}$. Let ϵ_i denote $\mathsf{NS}_\epsilon(h_i)$. Let C denote the constant from Theorem 3.10.4.

We begin by reducing to the case in which each q_i is not too small, and each $\epsilon_i \leq q_i$. Let $s_i \in \{+1, -1\}$ be the less likely value for $h_i(x)$, so s_i is such that $\mathbf{P}[h_i(x) = s_i] = q_i$. Let

$$\alpha = 2\sqrt{C}\ (\epsilon \ln k)^{\frac{1}{4}},$$

and let $S_\alpha = \{i \in [k] \colon q_i \leq \alpha/k\}$. Recalling that both x and $N_\epsilon(x)$ are both uniformly distributed, a union bound tells us that the probability there exists an $i \in S_\alpha$ such that $h(x) = s_i$ or $h(N_\epsilon(x)) = s_i$ is at most $\sum_{i \in S_\alpha} 2q_i \leq \sum_{i \in S_\alpha} 2\alpha/k \leq 2\alpha$. Since an additive $\alpha = O(1)(\epsilon \log k)^{\frac{1}{4}}$ does not affect the bound we are trying to prove, we may assume without loss of generality that $h_i(x) = h_i(N_\epsilon(x)) = -s_i$ for all $i \in S_\alpha$. In this case these halfspaces are irrelevant to the noise sensitivity calculation, and we can therefore assume without loss of generality that $S_\alpha = \emptyset$; i.e., that $q_i > \alpha/k$ for all i.

Next, note that each $\epsilon_i \leq C(2q_i)\left(\epsilon \ln \frac{1}{q_i}\right)^{\frac{1}{2}}$ by Theorem 3.10.4. We claim that we may assume $C(2q_i)\left(\epsilon \ln \frac{1}{q_i}\right)^{\frac{1}{2}} \leq q_i$ and hence $\epsilon_i \leq q_i$. The claimed inequality holds if and only if $q_i \geq \exp\left(-\frac{1}{4C^2\epsilon}\right)$; since each $q_i > \alpha/k = 2\sqrt{C}\epsilon^{\frac{1}{4}}\left(\frac{\ln^{\frac{1}{4}} k}{k}\right)$, it suffices to establish $2\sqrt{C}\ \epsilon^{\frac{1}{4}}\frac{\ln^{\frac{1}{4}} k}{k} \geq \exp\left(-\frac{1}{4C^2\epsilon}\right)$. It is easy to check that this holds so long as $k < \exp(\frac{1}{4C^2\epsilon})$ because $\frac{\ln^{\frac{1}{4}} k}{k}$ decreases for $k \geq 2$. But this holds without loss of generality, for otherwise $\epsilon \ln k \geq \frac{1}{4C^2} = \Omega(1)$ and the theorem is trivially true.

To summarize, we may assume without loss of generality,

$$q_i \geq \alpha/k, \qquad\qquad q_i \geq \epsilon_i. \tag{3.33}$$

Let F_i denote the random variable $h_i(N_\epsilon(x)) - h_i(x)$. Since

$$\mathbf{P}[h_i(x) = +1, h_i(N_\epsilon(x)) = -1] = \mathbf{P}[h_i(x) = -1, h_i(N_\epsilon(x)) = +1] = \epsilon_i/2,$$

we have that $F_i = \pm 2$ with probability $\epsilon_i/2$ each, and $F_i = 0$ with probability $1 - \epsilon_i$. Let H denote $\sum_{i=1}^k h_i(x)$ and let F denote $\sum_{i=1}^k F_i$; thus F denotes the amount by which $h_1(x) + \cdots + h_k(x)$ and $h_1(N_\epsilon(x)) + \cdots + h_k(N_\epsilon(x))$ differ. The proof of the present theorem centers around the observation that if $f(x) \neq f(N_\epsilon(x))$ then we must have $|F| \geq |H - \theta|$. Hence

$$\mathsf{NS}_\epsilon(f) \leq \mathbf{P}[|F| \geq |H - \theta|],$$

and we shall proceed by upper-bounding the probability that $|F| \geq |H - \theta|$.

While the random variables H and F are clearly not independent, each is a sum of independent Bernoulli random variables, and this allows us to bring to bear a number of standard estimates. Let us write σ_H^2 and σ_F^2 for the variances of H and F respectively. To upper-bound the probability that H is close to the threshold θ we shall use the following result, whose proof we defer to the end of this section:

Lemma 3.12.2 *Let $H_1, \ldots, H_k$ be independent ± 1-valued random variables, let $H = \sum_{i=1}^k H_i$, and let σ_H^2 denote* $\mathrm{Var}[H]$. *Then for every $\theta \in \mathbf{R}$ and every $\lambda \geq 1$ we have*

$$\mathbf{P}[|H - \theta| \leq \lambda] \leq O(1)\frac{\lambda}{\sigma_H}.$$

To bound the size of $|F|$ in terms of H's deviation from θ, we will use an estimate for

σ_F/σ_H. We claim

$$\sigma_F/\sigma_H \le O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}}. \tag{3.34}$$

To see this, let us compute,

$$\sigma_F^2 = \sum_{i=1}^{k} \mathrm{Var}[F_i] = 4\sum_{i=1}^{k} \epsilon_i \le 8C\sum_{i=1}^{k} q_i\left(\epsilon \ln \frac{1}{q_i}\right)^{\frac{1}{2}} \le O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{2}} \sum_{i=1}^{k} q_i,$$

where the first inequality follows from Theorem 3.10.4, and the second because $q_i \ge \alpha/k$ using (3.33). As for the variance of H,

$$\sigma_H^2 = \sum_{i=1}^{k} \mathrm{Var}[H_i(x)] = \sum_{i=1}^{k}(1-(1-2p_i)^2) = \sum_{i=1}^{k} 4p_i(1-p_i) = 4\sum_{i=1}^{k} 4q_i(1-q_i) \ge 2\sum_{i=1}^{k} q_i,$$

where the inequality uses $q_i \le \frac{1}{2}$. The inequality claimed in (3.34) now follows.

The proof now splits into two cases, depending on whether $\sigma_F \ge 1$ or $\sigma_F < 1$.

Case 1: $\boldsymbol{\sigma_F^2 \ge 1}$. In this case, we shall use a simple tail bound to show that F's magnitude is unlikely to exceed a moderate quantity times σ_F. Recall that F is the sum of the independent random variables F_i, and each one has mean zero and satisfies $|F_i| \le 2$. By Bernstein's inequality (a special case of Hoeffding's inequality; see, e.g., Section 2.2 of [Pet95]), for any $\tau > 0$ we have

$$\mathbf{P}[|F| \ge \tau\sigma_F] \le 2\exp\left[-\frac{(\tau\sigma_F)^2}{2(\sigma_F^2 + 2\tau\sigma_F)}\right] \le 2\exp\left[-\frac{\tau^2}{2+4\tau/\sigma_F}\right] \le 2\exp\left[-\frac{\tau^2}{2+4\tau}\right],$$

where the last inequality uses $\sigma_F \ge 1$.

As for H, we shall take $\tau \ge 1$, so $\tau\sigma_F \ge 1$ and we can apply Lemma 3.12.2 to conclude

$$\mathbf{P}[|H-\theta| \le \tau\sigma_F] \le O(1)\frac{\tau\sigma_F}{\sigma_H} \le O(1)\tau\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}},$$

where the second inequality is from (3.34). Hence

$$\begin{aligned}\mathbf{P}[|F| \ge |H-\theta|] &\le \mathbf{P}[|F| \ge \tau\sigma_F] + \mathbf{P}[|H-\theta| \le \tau\sigma_F] \\ &\le 2\exp\left[-\frac{\tau^2}{2+4\tau}\right] + O(1)\tau\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}}.\end{aligned}$$

Taking $\tau = A \log(1/(\epsilon \log \frac{k}{\alpha}))$ with A a suitably large constant we get

$$\mathsf{NS}_\epsilon(f) \le O(1) \left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}} \log(1/(\epsilon \log \tfrac{k}{\alpha})).$$

Case 2: $\boldsymbol{\sigma_F^2 < 1}$. In this case, let S denote the random set $\{i \colon h_i(x) \neq h_i(N_\epsilon(x))\}$; we refer to S as the *flip set.* We now consider the random variable H *conditioned on S being the flip set.* As conditional random variables, we have $(H|S) = \sum_{i=1}^k (h_i(x)|S)$, and the random variables $(h_i(x)|S)$ are still independent. It is easily verified that for $i \in S$, $(h_i(x)|S)$ takes the values ± 1 with equal probability; whereas, for $i \notin S$, $(h_i(x)|S) = -1$ with probability $\frac{p_i - \epsilon_i/2}{1-\epsilon_i}$ and $(h_i(x)|S) = +1$ with probability $\frac{1-p_i-\epsilon_i/2}{1-\epsilon_i}$. Thus for $i \in S$, $\mathrm{Var}[h_i(x)|S] = 1$, and for $i \notin S$,

$$\mathrm{Var}[h_i(x)|S] = 1 - \left(\frac{1-2p_i}{1-\epsilon_i}\right)^2 = 1 - \frac{(1-2q_i)^2}{(1-\epsilon_i)^2} \ge \frac{(1-2q_i)^2}{(1-q_i)^2} \ge 2q_i(1-q_i),$$

where the first inequality uses $\epsilon_i \le q_i$ from (3.33) and the second is elementary for $q_i \in [0, \frac{1}{2}]$. Recalling that $\mathrm{Var}[h_i(x)] = 4q_i(1-q_i)$ we see that regardless of whether $i \in S$ or not, $\mathrm{Var}[h_i(x)|S] \ge \frac{1}{2}\mathrm{Var}[h_i(x)]$. Hence $\mathrm{Var}[H|S] \ge \frac{1}{2}\sigma_H^2$. It thus follows from Lemma 3.12.2 that for every $\theta \in \mathbf{R}$ and $\lambda \ge 1$, $\mathbf{P}[|H-\theta| \le \lambda \mid S] \le O(1)\sqrt{2}\frac{\lambda}{\sigma_H}$, independently of S. Since $|F| \le 2|S|$ is immediate, we conclude,

$$\begin{aligned}
\mathsf{NS}_\epsilon(f) &\le \mathbf{P}[|H-\theta| \le 2|S|] \\
&= \sum_{S \subseteq [k]} \mathbf{P}[\text{flip set is } S] \cdot \mathbf{P}[|H-\theta| \le 2|S| \mid \text{flip set is } S] \\
&\le \sum_{S \subseteq [k]} \mathbf{P}[\text{flip set is } S] \cdot O(1)\frac{|S|}{\sigma_H} \\
&= O(1)\mathbf{E}[|S|]/\sigma_H \\
&= O(1)\left(\sum_{i=1}^k \epsilon_i\right)/\sigma_H \\
&= O(1)(\sigma_F^2/\sigma_H) \\
&\le O(1)(\sigma_F/\sigma_H),
\end{aligned}$$

where the last inequality is since $\sigma_F \leq 1$ in Case 2. Hence we have $\mathsf{NS}_\epsilon(f) \leq O(1)(\sigma_F/\sigma_H) \leq O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}}$ by (3.34).

In conclusion, in both Cases 1 and 2 we have

$$\mathsf{NS}_\epsilon(f) \leq O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}} \log(1/(\epsilon \log \tfrac{k}{\alpha})).$$

Since $\alpha = O(1)(\epsilon \ln k)^{\frac{1}{4}}$, the proof is complete. □

3.12.1 Proof of Lemma 3.12.2

In this subsection we prove the technical lemma needed in the preceding proof of Theorem 3.12.1:

Proposition 3.12.3 *Let $X_1, \ldots, X_n$ be independent ± 1-valued random variables where $\mathbf{P}[X_k = -1] = p_k$ and let $x = \sum_{k=1}^n X_k$. Then for every $\theta \in \mathbf{R}$,*

$$\mathbf{P}[|x - \theta| \leq 1] \leq \frac{O(1)}{\sqrt{\sum_{k=1}^n p_k(1-p_k)}}.$$

Lemma 3.12.2 as stated earlier follows easily from this via a union bound over a suitably chosen sequence of values for θ. The proof given below is based on similar arguments in Petrov's work [Pet95].

Proof: Define

$$p(x) = \frac{2(1-\cos x)}{x^2} \geq 0 \qquad \text{and} \qquad h(t) = \begin{cases} 1 - |t|, & |t| \leq 1 \\ 0, & \text{else} \end{cases}.$$

Elementary integration by parts shows that $p(x)$ is the inverse Fourier transform of $h(t)$; i.e.,

$$p(x) = \int_{-\infty}^{\infty} e^{-itx} h(t)\, dt.$$

By considering the Taylor expansion of $\cos x$, we get that $p(x) \geq \frac{11}{12}$ on the interval $[-1,1]$. Hence

$$\begin{aligned}
\mathbf{P}[|x-\theta| \leq 1] &= \mathop{\mathbf{E}}_{x}[\mathbf{1}_{x\in[\theta-1,\theta+1]}] \\
&\leq \tfrac{12}{11}\mathbf{E}[p(x-\theta)] \\
&= \tfrac{12}{11}\mathbf{E}\Big[\int_{-\infty}^{\infty} e^{-it(x-\theta)}h(t)\,dt\Big] \\
&= \frac{12}{11}\int_{-\infty}^{\infty} \mathbf{E}[e^{-itx}e^{it\theta}h(t)]\,dt \\
&= \frac{12}{11}\Big|\int_{-\infty}^{\infty} e^{it\theta}h(t)\mathbf{E}[e^{-itx}]\,dt\Big| && (3.35) \\
&\leq \frac{12}{11}\int_{-1}^{1}\big|\mathbf{E}[e^{-itx}]\big|\,dt, && (3.36)
\end{aligned}$$

with (3.35) following because the quantity is already real and nonnegative, and (3.36) following because $|e^{it\theta}| \leq 1$, $h(t) = 0$ outside $[-1,1]$, and $|h(t)| \leq 1$ otherwise.

Now,

$$\begin{aligned}
\mathop{\mathbf{E}}_{x}[e^{-itx}] &= \mathop{\mathbf{E}}_{x_k \leftarrow X_k}\left[\exp(-it\sum_{k=1}^{n} x_k)\right] \\
&= \mathop{\mathbf{E}}_{x_k \leftarrow X_k}\Big[\prod_{k=1}^{n}\exp(-itx_k)\Big] \\
&= \prod_{k=1}^{n}\mathop{\mathbf{E}}_{x_k \leftarrow X_k}[\exp(-itx_k)] \qquad \text{(by independence)} \\
&= \prod_{k=1}^{n}(p_k\exp(it) + (1-p_k)\exp(-it)) \\
&= \prod_{k=1}^{n}(\cos t + i(2p_k-1)\sin t).
\end{aligned}$$

By comparing Taylor expansions one can establish that

$$|\cos t + i(2p-1)\sin t| \leq \exp\Big(-\frac{11}{24}p(1-p)t^2\Big)$$

for $p \in [0,1], t \in [-1,1]$. We may conclude the following:

$$\begin{aligned}
\mathbf{P}[|x-\theta| \leq 1] &\leq \frac{12}{11}\int_{-1}^{1}\prod_{k=1}^{n}\exp(-\tfrac{11}{24}p_k(1-p_k)t^2)\,dt \\
&= \frac{12}{11}\int_{-1}^{1}\exp\left(-\frac{11}{24}\left[\sum_{k=1}^{n}p_k(1-p_k)\right]t^2\right)\,dt \\
&= \frac{12}{11}\int_{-1}^{1}\exp\left(-\frac{t^2}{2\left(\sqrt{\frac{12}{11}}\left(\sum p_k(1-p_k)\right)^{-1/2}\right)^2}\right)\,dt \\
&\leq \frac{12}{11}\int_{-\infty}^{\infty}\exp\left(-\frac{t^2}{2\left(\sqrt{\frac{12}{11}}\left(\sum p_k(1-p_k)\right)^{-1/2}\right)^2}\right)\,dt \\
&= \sqrt{2\pi}(\tfrac{12}{11})^{3/2}\left(\sum p_k(1-p_k)\right)^{-1/2},
\end{aligned}$$

since $(\sqrt{2\pi}\sigma)^{-1}\exp(-t^2/2\sigma^2)$ is a probability density function for every positive σ. □

Chapter 4

Hardness Amplification

In this chapter we relate noise sensitivity to the problem of *hardness amplification* from computational complexity theory. In hardness amplification, the goal is to take a boolean function f assumed to be slightly hard to compute and to produce a new function h which is much harder to compute. Here we mean "hardness" in the sense of hardness of average; we consider the fraction of inputs on which the function is computed correctly:

Definition 4.0.4 *We say a boolean function* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is* $(1-\epsilon)$-hard for circuits of size s *if no circuit of size* s *can correctly compute* f *on a* $1-\epsilon$ *fraction of the inputs* $\{+1,-1\}^n$.

The most straightforward way to perform hardness amplification is to use what is called a "direct product" theorem (see, e.g., Shaltiel [Sha01] for a study of direct product theorems). Such a theorem quantifies the hardness of a composite function $g \otimes f$ in terms of an assumed hardness of f and some intrinsic property of g. Prior to this work, the only known direct product theorem was the classical Yao XOR Lemma (originating in [Yao82]) which dealt with the case $g = \mathsf{PARITY}_k$. Roughly speaking, the Yao XOR Lemma says that if f is $(1-\epsilon)$-hard to compute then $\mathsf{PARITY}_k \otimes f$ is $(\frac{1}{2} + \frac{1}{2}(1-2\epsilon)^k)$-hard to compute. As we shall see, it is not a coincidence that this quantity is precisely $1 - \mathsf{NS}_\epsilon(\mathsf{PARITY}_k)$.

In this chapter we prove a nearly sharp direct product theorem for every possible g. That is, we answer the following question:

Question 4.0.5 *Suppose* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is a balanced boolean function which is* $(1-\epsilon)$*-hard for circuits of size* s*. Let* $g\colon \{+1,-1\}^k \to \{+1,-1\}$ *be any function. What is the hardness of the composite function* $g \otimes f$*?*

(We consider only balanced functions f for technical reasons that will become clear later. In fact, our final theorem will allow for slightly imbalanced functions.)

The near-sharp answer we give to the above question is in terms of an intrinsic property of g, parameterized by ϵ, which we call the *expected bias of g*. This quantity is very closely related to the noise sensitivity of g as we shall see. Let us now define expected bias:

Definition 4.0.6 *The* bias *of a boolean function h is* $\mathrm{bias}(h) = \max\{\mathbf{P}[h = \mathrm{T}], \mathbf{P}[h = \mathrm{F}]\}$.

Definition 4.0.7 *We denote by R_ϵ^n the set of random restrictions ρ on n coordinates, in which each coordinate is mapped independently to $\star$ with probability ϵ, to 0 with probability $(1-\epsilon)/2$, and to 1 with probability $(1-\epsilon)/2$. Given a boolean function $h\colon \{+1,-1\}^n \to \{+1,-1\}$, we write h_ρ for the function given by applying restriction ρ to function h.*

Definition 4.0.8 *Given a boolean function $h\colon \{+1,-1\}^n \to \{+1,-1\}$ and $0 \le \epsilon \le 1$, the* expected bias of h at ϵ *is*

$$\mathsf{EB}_\epsilon(h) = \mathop{\mathbf{E}}_{\rho \in R_\epsilon^n}[\mathrm{bias}(h_\rho)].$$

With this definition in place, we can now answer Question 4.0.5:

Answer 4.0.5 *Roughly speaking, the hardness of $g \otimes f$ is $\mathsf{EB}_{2\epsilon}(g)$. To state our direct product theorem exactly,*

Theorem 4.0.9 *For every $\delta > 0$, the function $g \otimes f$ is $(\mathsf{EB}_{(2-\delta)\epsilon}(g) + \eta)$-hard for circuits of size $s' = \Omega(\frac{\eta^2/\log(1/\epsilon)}{k}\, s)$.*

In Section 4.2 we explain from an intuitive, information-theoretic perspective why Answer 4.0.5 should be the correct answer; the proof of Theorem 4.0.9 in Section 4.3 transfers the ideas to the computational setting of circuits. We also use the intuitive ideas and constructions of Shaltiel to show that Theorem 4.0.9 is nearly tight; see Section 4.6.

Note that we get a form of Yao's XOR Lemma as an immediate corollary of Theorem 4.0.9; since $\mathsf{EB}_\epsilon(\mathsf{PARITY}_k)$ is easily calculated to be $\frac{1}{2} + \frac{1}{2}(1-\epsilon)^k$, we get the following (taking $\delta = 1$):

Corollary 4.0.10 *If f is a balanced boolean function which is $(1-\epsilon)$-hard for circuits of size s, then $f \oplus f \oplus \cdots \oplus f$ (k times) is $(\frac{1}{2} + \frac{1}{2}(1-\epsilon)^k + \eta)$-hard for circuits of size $\Omega(\frac{\eta^2/\log(1/\epsilon)}{k}\, s)$.*

As a practical matter, it is often quite difficult to calculate the expected bias of even relatively simple functions. Fortunately, noise stability (i.e., $1 - \mathsf{NS}_\epsilon$) is an excellent estimator for expected bias. This allows us to put to use the calculations from Chapter 3. Qualitatively, we have that $\mathsf{EB}_{2\epsilon}(g) = 1 - o(1)$ if and only if $1 - \mathsf{NS}_\epsilon(g) = 1 - o(1)$, and the same holds for $1 - \Omega(1)$ and $\frac{1}{2} + o(1)$ in place of $1 - o(1)$. Quantitatively, we have the following:

Proposition 4.0.11 *Let g be a boolean function and let $\epsilon \in [0, \frac{1}{2}]$. Write $\gamma = 2(\frac{1}{2} - \mathsf{NS}_\epsilon(g))$. Then*

$$1 - \mathsf{NS}_\epsilon(g) = \frac{1}{2} + \frac{1}{2}\gamma \quad \leq \quad \mathsf{EB}_{2\epsilon}(g) \quad \leq \quad \frac{1}{2} + \frac{1}{2}\gamma^{\frac{1}{2}}.$$

Proof: Given a $\rho \in R^n_\epsilon$, write stars(ρ) for the set of coordinates to which ρ assigns $\star$. By Fact 2.1.6 we have $\gamma = \mathbf{E}_{x, y = N_\epsilon(x)}[g(x)g(y)]$. An equivalent way of generating x and y is to first pick a random restriction $\rho \in R^n_{2\epsilon}$ and then to pick two independent, uniformly random strings x' and y' for the coordinates stars(ρ). Therefore

$$\gamma = \mathop{\mathbf{E}}_{\substack{\rho \in R^n_{2\epsilon} \\ x', y' \in \{+1,-1\}^{\mathrm{stars}(\rho)}}} [g_\rho(x')g_\rho(y')] = \mathbf{E}_\rho\left[\mathbf{E}_{x'}[g_\rho(x')]\mathbf{E}_{y'}[g_\rho(y')]\right] = \mathbf{E}_\rho\left[\mathbf{E}[g_\rho]^2\right].$$

On the other hand, using $\mathrm{bias}(g_\rho) = \frac{1}{2} + \frac{1}{2}|\mathbf{E}[g_\rho]|$ and linearity of expectation, we get $\mathsf{EB}_{2\epsilon}(g) = \frac{1}{2} + \frac{1}{2}\mathbf{E}_{\rho \in R^n_{2\epsilon}}[|\mathbf{E}[g_\rho]|]$. Thus to complete the proof we need only show that $\gamma \leq \mathbf{E}_\rho[|\mathbf{E}[g_\rho]|] \leq \gamma^{\frac{1}{2}}$, where $\gamma = \mathbf{E}_\rho\left[\mathbf{E}[g_\rho]^2\right]$. But this follows immediately from Cauchy-Schwarz and the fact that $|\mathbf{E}[g_\rho]| \leq 1$. □

We briefly note that the bounds in Proposition 4.0.11 cannot be much improved. As we have seen, the lower bound becomes equality for $g = \mathsf{PARITY}_k$ and every $0 \leq \epsilon \leq \frac{1}{2}$. As for the upper bound, a very easy modification of the proof of Theorem 3.4.2 establishes the following:

Theorem 4.0.12 *For any $\epsilon \in [0, 1]$,*

$$\lim_{n \to \infty} \mathsf{EB}_\epsilon(\mathsf{MAJ}_n) = \frac{3}{4} + \frac{1}{2\pi}\arcsin(1 - 2\epsilon).$$

(The error estimate is also the same as in Theorem 3.4.2.)

Consider now $\mathsf{NS}_{\frac{1}{2}-\delta}(\mathsf{MAJ}_n)$ and $\mathsf{EB}_{1-2\delta}(\mathsf{MAJ}_n)$ for small δ and very large n. Using Theorems 3.4.2 and 4.0.12 and the approximation from Fact 3.4.3, we get

$$\frac{1}{2}+\frac{1}{2}\gamma \approx \frac{1}{2}+(2/\pi)\delta \qquad \mathsf{EB}_{1-2\delta} \approx \frac{1}{2}+(\sqrt{2}/\pi)\delta^{\frac{1}{2}} \qquad \frac{1}{2}+\frac{1}{2}\gamma^{\frac{1}{2}} \approx \frac{1}{2}+(1/\sqrt{\pi})\delta^{\frac{1}{2}}.$$

So in this case the upper bound in Proposition 4.0.11 is asymptotically tight.

4.1 Motivation: the hardness of NP

Our main motivation for studying hardness amplification is to try to quantify the hardness on average of complexity classes — especially NP. That is, we would like to know how hard the hardest function in NP is for polynomial-sized circuits. Of course, we do not know if 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 question. However, under the reasonable assumption that NP is at least slightly hard on average we can use hardness amplification results to quantify 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, e.g., [BFNW93, Imp95, IW97, STV01]. Specifically, it is known that if EXP is $(1-2^{-n})$-hard for polynomial circuits — i.e., if $\mathsf{EXP} \not\subseteq \mathsf{P}$/poly — then EXP contains a function which is $(\frac{1}{2}+1/\text{poly}(n))$-hard for polynomial circuits. A crucial ingredient in some of the proofs involved is the Yao XOR Lemma. Unfortunately, the XOR Lemma cannot be used to amplify hardness within NP. The reason is simple: $\mathsf{PARITY}_k \otimes f$ is not necessarily in NP, even when f is. Given $f \in \mathsf{NP}$, the only easy way to ensure that $g \otimes f$ is also in NP is to take g to be *monotone* (and also in NP). Thus we are naturally led to prove hardness amplification results for monotone g, and this leads us to look for monotone functions which are very noise-sensitive. This motivated our work in Sections 3.6, 3.7, 3.8, 3.9.

In addition to Theorem 4.0.9, the main result we prove in this chapter is a hardness theorem for NP:

Theorem 4.1.1 *If there is a function family in* NP *which is infinitely often balanced and* $(1 - 1/\text{poly}(n))$*-hard for circuits of polynomial size, then for any constant* $\gamma > 0$ *there is a function family in* NP *which is infinitely often* $(\frac{1}{2} + n^{-\frac{1}{2}+\gamma})$*-hard for circuits of polynomial size. (Here "infinitely often" simply means for infinitely many input lengths* n*.)*

As we will see in Section 4.5, $\frac{1}{2}+n^{-\frac{1}{2}+\gamma}$ is nearly optimal for our techniques; getting hardness down to, say, $\frac{1}{2} + n^{-1}$ would require a significant departure, if indeed this is even possible.

The technical assumption in this theorem that the initial hard function is balanced may be removed at the expense of a small loss in final hardness:

Theorem 4.1.2 *If there is a function family in* NP *which is infinitely often* $(1-1/\text{poly}(n))$*-hard for circuits of polynomial size, then for any constant* $\gamma > 0$ *there is a function family in* NP *which is infinitely often* $(\frac{1}{2} + n^{-\frac{1}{3}+\gamma})$*-hard for circuits of polynomial size.*

4.2 Intuition for the direct product theorem

In this section we explain the intuition behind the hardness bound given in Answer 4.0.5. Suppose that $f \colon \{+1,-1\}^n \to \{+1,-1\}$ is a balanced function which is $(1-\epsilon)$-hard to compute, and let $g \colon \{+1,-1\}^k \to \{+1,-1\}$ be the hardness amplifying function. Let us try to understand what the hardness of computing $g \otimes f$ should be, and how it relates to the expected bias and noise sensitivity of g.

Suppose we are trying to compute $g \otimes f$ using modest computational resources. Inputs $w_1, \dots, w_k \in \{+1,-1\}^n$ are selected independently and uniformly at random and our task is to compute $g(f(w_1), \dots, f(w_k))$. Let us write $x_i = f(w_i)$ and call these the "true inputs" to g. Since f is balanced and the w_i's are independent, the true inputs x_i are simply independent uniform random bits.

Here is the most naive strategy we could use for trying to compute $(g \otimes f)(w_1, \dots, w_k) = g(x_1, \dots, x_k)$: First we try to compute each x_i; say our computation for $f(w_i)$ produces the possibly mistaken "guess input" y_i. We then simply guess $g(y_1, \dots, y_k)$. Since f is $(1-\epsilon)$-hard, each guess input y_i will be equal to x_i with probability $1-\epsilon$. Thus our probability of success — the probability that $g(y_1, \dots, y_k) = g(x_1, \dots, x_k)$ — is precisely $1 - \mathsf{NS}_\epsilon(g)$. We conclude that it is reasonable for a small circuit to compute $g \otimes f$ with probability $1 - \mathsf{NS}_\epsilon(g)$. Indeed, as we see from Proposition 4.0.11, $1 - \mathsf{NS}_\epsilon(g) \leq \mathsf{EB}_{2\epsilon}(g)$, the hardness bound we are claiming.

However there are situations in which strategies better than the naive one are possible; in order to definitively say that $g \otimes f$ has a certain hardness, we must consider the best possible strategy. It is easy to see that $1 - \mathsf{NS}_\epsilon(g)$ will not be the best possible bound. For example, given the guess inputs y_i, guessing $g(y_1, \ldots, y_n)$ is not necessarily the best idea. Suppose $g = \mathsf{AND}_k$ and ϵ is quite large. Then even if we compute $y_1 = y_2 = \cdots = y_k = \mathrm{T}$, knowing ϵ it is still more probable than not that at least one x_i equals F. Thus our best strategy would be to always guess F. In general, if all we know about the y_i's is that they are correct with probability $1 - \epsilon$, the best strategy is a maximum likelihood one, computed by taking a weighted average of g's values near $(y_1, \ldots, y_k)$.

But even this might not be the best strategy for us, since it neglects a crucial possibility: we might have extra information as to whether or not y_i is correct. One could imagine that for each input w_i, it is easy to tell how hard computing $f(w_i)$ is, and thus how confident we should be in our guess y_i. Naturally, this extra confidence information should influence our strategy for guessing $g(x_1, \ldots, x_k)$. Since f is $(1 - \epsilon)$-hard, the *average* confidence we achieve cannot exceed $1 - \epsilon$. But it is not hard to see that it is better for us to know most inputs with confidence 1 rather than for us to know each input with confidence $1 - \epsilon$. Let us give a name to this more helpful scenario:

Definition 4.2.1 *In the* Hard-Core scenario *(the name is inspired by [Imp95]), for each i, with probability* $1 - 2\epsilon$ *our guess* y_i *is the correct value of* x_i *and furthermore* we know that our guess is correct*; with probability* 2ϵ, y_i *is a uniformly random bit and* we know that this guess input is completely uncorrelated to x_i*.*

Note that in the Hard-Core scenario, each guess bit y_i is still correct (equal to x_i) with probability $1 - \epsilon$. To see that this scenario is more helpful than the situation in the naive strategy, simply note that we can simulate the latter by ignoring the extra confidence information.

The Hard-Core scenario in some sense gives us the most information — i.e., it models f as being $(1 - \epsilon)$-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 y_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. That is, if the function is biased towards $+1$, guess $+1$; 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 2ϵ — i.e., $\mathsf{EB}_{2\epsilon}(g)$.

It turns out that using this optimal strategy in the Hard-Core scenario is the best we can ever do — we show essentially $\mathsf{EB}_{2\epsilon}(g)$ hardness for $g \otimes f$ in Theorem 4.0.9. Further, as we will see in Theorem 4.6.7, it is perfectly possible for f to be hard in the manner of the Hard-Core scenario; i.e., for f to be trivial on an easily recognized $1 - 2\epsilon$ fraction of inputs and nearly impossible to compute on a 2ϵ fraction. In this case, assuming that g is not too complicated a function, we can actually perform the best guessing strategy and thus Theorem 4.0.9 is nearly tight.

4.3 The hardness theorem

In this section we give the computational proof of Theorem 4.0.9, 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 $(1-\epsilon)$-hard function conforms to the Hard-Core scenario described in Section 4.2. That is, on a 2ϵ-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 4.3.1 *Let f be $(1-\epsilon)$-hard for size s. Then for every constant $\delta > 0$, f has a "hard-core" $S \subseteq \{+1,-1\}^n$ of size at least $(2-\delta)\epsilon 2^n$ and at most $2\epsilon 2^n$ with the following property: on S, f is balanced and $(\frac{1}{2}+\eta)$-hard for size $\Omega(\frac{\eta^2}{\log(1/\epsilon)}\, s)$.*

Proof: Combining the results of Impagliazzo [Imp95] with the improvements of Klivans and Servedio [KS03] (see also [Ser01]), there must exist a set $S' \subseteq \{+1,-1\}^n$ of size at least $(2-\delta)\epsilon 2^n$ and at most $(2-\delta/2)\epsilon 2^n$ on which f is $(\frac{1}{2}+\eta/2)$-hard for size $\Omega(\frac{\eta^2}{\log(1/\epsilon)}\, s)$. We would like to add a small number of strings to S' to make f balanced on this set. Suppose without loss of generality that f is biased towards $+1$ on S'. Since f is $(\frac{1}{2}+\eta/2)$-hard on S', it is $+1$ on at most $(\frac{1}{2}+\eta/2)|S'|$ strings in S'. Since $|S'| \leq (2-\delta/2)\epsilon 2^n$ and we may assume without loss of generality that $\eta < \delta/4$, the total number of $+1$'s f has in S' is at most $(\frac{1}{2}+\eta/2)(2-2\eta)\epsilon 2^n \leq \epsilon 2^n$.

On the other hand, since f is $(1-\epsilon)$-hard on $\{+1,-1\}^n$, f must take on the value -1 for at least $\epsilon 2^n$ strings in $\{+1,-1\}^n$. It follows that there is a superset $S' \subseteq S \subseteq \{+1,-1\}^n$

of S' on which f is balanced. This set has size at most $2\epsilon 2^n$.

It remains to show that f is $(\frac{1}{2}+\eta)$-hard on S. We know that no circuit (of size $\Omega(\frac{\eta^2}{\log(1/\epsilon)}\,s)$) gets f right on S' with probability more than $\frac{1}{2}+\eta/2$, and no circuit gets f right on $S \setminus S'$ with probability more than 1. We know that

$$|S \setminus S'| = \text{bias}(f|_{S'})|S'| - (1-\text{bias}(f|_{S'}))|S'| = (2\text{bias}(f|_{S'})-1)|S'| \leq \eta|S'|.$$

It follows that no circuit gets f right on S with probability more than

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

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 4.2. 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 4.3.2 *Let $\mathcal{H}_n$ denote the n-dimensional Hamming cube with its graph structure, and suppose $h\colon \mathcal{H}_n \to \{+1,-1\}$ and $p\colon \mathcal{H}_n \to [0,1]$. Further suppose that*

$$2^{-n}\left[\sum_{x\in h^{-1}(+1)} p(x) + \sum_{x\in h^{-1}(-1)} (1-p(x))\right] \tag{4.1}$$

is at least $\text{bias}(h)+\eta$*. Then there exists an edge (x,y) in $\mathcal{H}_n$ with $|p(x)-p(y)| \geq \Omega(\eta/n^{\frac{1}{2}})$.*

Proof: The idea here is that h is the function whose value on $\{+1,-1\}^n$ we are trying to guess, and $p(x)$ represents the probability with which we guess $+1$ when the input is x. Our success probability is given by (4.1), and the lemma says that if we are doing strictly better than 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 $\Omega(\eta/n^{\frac{1}{2}})$ is a little tricky, so we defer the proof to Section 4.4. For now we just prove a lower bound of η/n. Note that using this weaker lower bound in the proof of Theorem 4.0.9 will make no qualitative difference, and will still be able to prove the hardness theorem for NP, Theorem 4.1.1, just as well.

Without loss of generality we may assume that h is biased towards $+1$; write $b = \mathrm{bias}(h) = \mathbf{P}[h = +1] \geq \frac{1}{2}$. By way of contradiction, assume $|p(x) - p(y)| < \eta/n$ for all edges $(x, y) \in \mathcal{H}_n$. Let M and m be the maximum and minimum values of p on $\mathcal{H}_n$. Since any of pair of points in the cube is at Hamming distance at most n, it follows that $M - m < \eta$. Now (4.1) would be maximized if $p(x) = M$ for all $x \in h^{-1}(+1)$ and $p(x) = m$ for all $x \in h^{-1}(-1)$. Hence

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

since $b \geq \frac{1}{2}$. This contradiction completes the proof. □

Now we prove our hardness amplification theorem. We will actually prove a slightly stronger statement than the one given in Section 4 — for technical reasons we want the theorem to hold under the assumption that the hard function f is only nearly balanced, not necessarily exactly balanced.

Theorem 4.0.9 *Let* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *be a function which is* $(1-\epsilon)$*-hard for size* s. *Let* $g\colon \{+1,-1\}^k \to \{+1,-1\}$ *be any function. Let* $\eta > 0$ *be any parameter, and assume that* $\mathrm{bias}(f) \leq \frac{1}{2} + (1-2\epsilon)\eta/4k$. *Then for every* $\delta > 0$, $g \otimes f$ *is* $(\mathsf{EB}_{(2-\delta)\epsilon}(g) + \eta)$*-hard for circuits of size* $s' = \Omega(\frac{\eta^2/\log(1/\epsilon)}{k}\, s)$.

Proof: Let S be the hard-core for f given by Lemma 4.3.1, using parameters ϵ, δ, and $\eta' = (c/8)\eta k^{-\frac{1}{2}}$, where c is the constant hidden in the $\Omega(\cdot)$ of Lemma 4.3.2. Then f is $(\frac{1}{2} + \eta')$-hard on S for circuits of size s'. Furthermore, we have $\mathrm{bias}(f|_S) = \frac{1}{2}$, and because $|S| \leq 2\epsilon 2^n$ and $\mathrm{bias}(f) \leq \frac{1}{2} + (1-2\epsilon)\eta/4k$, we get $\mathrm{bias}(f|_{S^c}) \leq \frac{1}{2} + \eta/4k$.

Let $\gamma = |S|2^{-n} \geq (2-\delta)\epsilon$, and let $E = \mathsf{EB}_\gamma(g)$. It's easy to see that $\mathsf{EB}_\epsilon(h)$ is a nonincreasing function of ϵ for any h — simply note that in the guessing game of Section 4.2, the more information the optimal strategy has the better. Hence $E \leq \mathsf{EB}_{(2-\delta)\epsilon}(g)$.

Suppose by way of contradiction that C is a circuit of size s' computing $g \otimes f$ on an $(\mathsf{EB}_{(2-\delta)\epsilon}(g) + \eta) \geq E + \eta$ fraction of the inputs in $(\{+1,-1\}^n)^k$. For a given restriction $\rho \in R^k_\gamma$, let us say that an input $(x_1, \dots, x_k) \in (\{+1,-1\}^n)^k$ *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 R^k_γ. In particular: the probability that $x_i \in S$ is exactly γ, which is the correct $\star$ probability; the probability that $f(x_i) = +1$ given that $x_i \in S^c$ is at most $\frac{1}{2} + \eta/4k$ because $\text{bias}(f|_{S^c}) \leq \frac{1}{2} + \eta/4k$; and, a similar statement holds for $f(x_i) = -1$. Hence $\mathbf{P}[(x_1,\dots,x_k) \text{ matches } \rho] \leq \mathbf{P}[\rho](\frac{1}{2}+\eta/4k)^k/(\frac{1}{2})^k \leq \mathbf{P}[\rho](1+\frac{2}{3}\eta)$ for every ρ.

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

$$\begin{array}{rrcl} & \mathbf{P}[C \text{ correct}] & \geq & E + \eta \\ \Rightarrow & \sum_\rho \mathbf{P}[(x_1,\dots,x_k) \text{ matches } \rho]c_\rho & \geq & \sum_\rho \mathbf{P}[\rho]\text{bias}(g_\rho) + \eta \\ \Rightarrow & \sum_\rho \mathbf{P}[\rho](1+\frac{2}{3}\eta)c_\rho & \geq & \sum_\rho \mathbf{P}[\rho]\text{bias}(g_\rho) + \eta, \end{array}$$

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

By reordering the k length-n inputs to C we may that assume ρ is of the form $(\underbrace{\star,\dots,\star}_{k' \text{ times}}, b_{k'+1},\dots,b_k)$ for some $k' < k$, $b_i \in \{+1,-1\}$. An averaging argument tells us there exist particular $x^*_{k'+1},\dots,x^*_k \in \{+1,-1\}^n$ 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 $\{+1,-1\}^n$, and it correctly computes g_ρ with probability at least c_ρ when its inputs $x_1,\dots,x_{k'}$ 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 $\{+1,-1\}^n$).

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

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

Since this quantity is at least $c_\rho \geq \mathrm{bias}(g_\rho) + \eta/4$, Lemma 4.3.2 tells us that there is an edge (z,z') in $\mathcal{H}_{k'}$ such that $|p(z)-p(z')| \geq c(\eta/4)(k')^{-\frac{1}{2}} \geq (c/4)\eta k^{-\frac{1}{2}} = 2\eta'$.

Reorder inputs again so that we may assume $z = (+1,u)$, $z' = (-1,u)$ for some string $u \in \{+1,-1\}^{k'-1}$. 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

$$\left|\mathbf{P}_{x_1\in(f|_S)^{-1}(+1)}[C'(x_1,x_2^*,\ldots,x_{k'}^*) = +1]\ -\mathbf{P}_{x_1\in(f|_S)^{-1}(-1)}[C'(x_1,x_2^*,\ldots,x_{k'}^*) = +1]\right| \geq 2\eta'.$$

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

$$\left|\mathbf{P}_{x_1\in(f|_S)^{-1}(+1)}[C''(x_1) = +1]\ -\mathbf{P}_{x_1\in(f|_S)^{-1}(-1)}[C''(x) = +1]\right| \geq 2\eta',$$

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

In Section 4.6 we give unconditional constructions showing that this theorem is nearly tight.

4.4 Proof of Lemma 4.3.2

In this section we prove the strong version of Lemma 4.3.2. The result is a simple example of the concentration of measure phenomenon. We shall slightly generalize by allowing p's range to be $[-1,1]$ rather than $[0,1]$.

Lemma 4.3.2 *Let $\mathcal{H}_n$ denote the n-dimensional Hamming cube with its graph structure, and suppose $h\colon \mathcal{H}_n \to \{+1,-1\}$ and $p\colon \mathcal{H}_n \to [-1,1]$. Further suppose $|\mathbf{E}[hp]| \geq |\mathbf{E}[h]| + \eta$. Then there exists an edge (x,y) in $\mathcal{H}_n$ such that $|p(x)-p(y)| \geq \Omega(\eta/n^{\frac{1}{2}})$.*

Proof: Let γ be the maximum of $|p(x)-p(y)|$ over all edges (x,y) in $\mathcal{H}_n$. For a subset $C \subseteq \mathcal{H}_n$, let $\mu_p(C)$ denote the average value of p on C. Let m be the median value of p on

$\mathcal{H}_n$, and partition $\mathcal{H}_n$ 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 $\mathcal{H}_n$ 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 $\mathcal{H}_n$ to a ball of radius $n/2$ is $O(n^{\frac{1}{2}})$; this fact is fairly standard. To see it, suppose without loss of generality the ball is $A = \{x \in \mathcal{H}_n \colon \mathsf{MAJ}(x) = +1\}$. Then the distance from a point x to A is equal to 0 if $\mathsf{MAJ}(x) = +1$, and is equal to $(\text{weight}(x) - n/2)$ otherwise. The result now follows from the fact that the distribution of weight(x) is approximately normal, $N(n/2, n/4)$, and that $\mathbf{E}[|N(n/2,n/4) - n/2|] = O(n^{\frac{1}{2}})$. Hence the average distance to A^+ over all points in $\mathcal{H}_n$ is $O(n^{\frac{1}{2}})$. 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 $O(n^{\frac{1}{2}})$ (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 certainly $|p(x) - p(y)| \leq d\gamma$. Since the smallest possible value of p on A^+ is m, it follows that $\mu_p(C) \geq m - O(\gamma n^{\frac{1}{2}})$ for any set $|C| = \frac{1}{2}2^n$. Running the same argument with A^- in place of A^+ yields that $\mu_p(C) \leq m + O(\gamma n^{\frac{1}{2}})$ for any set $|C| = \frac{1}{2}2^n$. Writing $\theta = O(\gamma n^{\frac{1}{2}})$, 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 $b = 2^{-n}|H^+| \geq \frac{1}{2}$. 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 $|\mathbf{E}[hp]| \geq |\mathbf{E}| + \eta$ by assumption. Hence $\theta \geq \eta$ which implies $\gamma \geq \Omega(\eta/n^{\frac{1}{2}})$. □

4.5 Hardness amplification within NP

As we mentioned earlier, we can use the hardness amplification theorem to show that if NP is slightly hard on average for polynomial circuits, then it is in fact very hard. In this section we shall prove Theorems 4.1.1 and 4.1.2. Our operating assumption about the hardness of NP will be that there is a function family (f_n) in NP which is infinitely often $(1-1/\text{poly}(n))$-hard for polynomial circuits. Our methods are more straightforward if this function family is also assumed to be balanced. However, this assumption appears restrictive; it is not clear if there is an NP-complete language which is balanced on all input lengths. We will use a pair of simple tricks to deal with the unbalanced case.

Let us first construct the monotone function we will use to amplify hardness:

Theorem 4.5.1 *For every $\alpha > 0$ there is a monotone function family (g_k), computable in* P *and defined for every input length k, such that $\mathsf{NS}_{1/2k^\alpha}(g_k) \geq \frac{1}{2} - \frac{1}{2}k^{-1+4\alpha}$ for all sufficiently large k.*

Proof: Let $\alpha' = 3\alpha$. On input length k, pick ℓ as large as possible so that $3^\ell \leq k^{\alpha'}$, and pick k' to be the largest possible input length less than $k^{1-\alpha'}$ for the functions T constructed in Theorem 3.9.8. Define $g_k = T_{k'} \otimes \mathsf{MAJ}_3^{\otimes \ell}$ to be on k bits by ignoring some input bits if necessary. Note that $3^\ell = \Omega(k^{\alpha'})$ and $k' = \Omega(k^{1-\alpha'})$, so we haven't lost too much in the input length. Furthermore, being tribes functions composed with recursive majorities of 3, the family (g_k) is computable in P.

Assume k is sufficiently large. Then by choice of ℓ and α', we have $1/2k^\alpha \geq 1/(3^\ell)^{\log_{\frac{3}{2}} 3}$. Thus by Theorem 3.7.2, $\mathsf{NS}_{1/2k^\alpha}(\mathsf{MAJ}_3^{\otimes \ell}) \geq \epsilon$ for some constant $\epsilon \geq \Omega(1)$. Next, from Theorem 3.9.8, $\mathsf{NS}_\epsilon(T_{k'}) \geq \frac{1}{2} - \frac{1}{2}(k')^{-1+\alpha} \geq \frac{1}{2} - \frac{1}{2}k^{-1+\alpha+\alpha'}$ for k sufficiently large. Therefore $\mathsf{NS}_{1/2k^\alpha}(g_k) \geq \frac{1}{2} - \frac{1}{2}k^{-1+4\alpha}$ by Proposition 2.2.7, as desired. □

Corollary 4.5.2 *The function family (g_k) from Theorem 4.5.1 satisfies*

$$\mathsf{EB}_{k^\alpha}(g_k) \geq \frac{1}{2} + \frac{1}{2}k^{-\frac{1}{2}+2\alpha}.$$

Proof: This follows immediately from Proposition 4.0.11. □

We can now prove Theorem 4.1.1:

Theorem 4.1.1 *If there is a function family (f_n) in NP which is infinitely often balanced and $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then for any constant $\gamma > 0$ there is a function family (h_m) in NP which is infinitely often $(\frac{1}{2} + m^{-\frac{1}{2}+\gamma})$-hard for circuits of polynomial size.*

Proof: Let γ be given and assume that (f_n) is in NP and is infinitely often balanced and $(1 - 1/n^c)$-hard for circuits of polynomial size. For each n, define $k = k(n) = n^{4c/\gamma}$. Write $m = kn$ and define $h_m = g_k \otimes f_n$, where g_k is the function from Theorem 4.5.1. Since g_k is monotone and in P, the family (h_m) is in NP. Now apply the hardness amplification theorem, Theorem 4.0.9, with $\delta = 1$, $\epsilon = 1/n^c$ and $\eta = 1/m$. Since all the parameters are fixed polynomials in n, we conclude that h_m is $(\mathsf{EB}_{1/n^c}(g_k) + 1/m)$-hard for polynomial circuits. By Corollary 4.5.2, $\mathsf{EB}_{1/n^c}(g_k) = \mathsf{EB}_{1/k^{\gamma/4}}(g_k) \leq \frac{1}{2} + \frac{1}{2}k^{-\frac{1}{2}+\gamma/2}$. But $m = k^{(\gamma/4c)/(1+\gamma/4c)}$, so we may easily conclude that h_m is $(\frac{1}{2} + \frac{1}{2}m^{-\frac{1}{2}+\gamma} + 1/m)$-hard for polynomial circuits, for m sufficiently large. This completes the proof. □

Let us now remark that this bound of $\frac{1}{2} + m^{-\frac{1}{2}+\gamma}$ is close to the best we can achieve for NP using our hardness amplification technique. By Proposition 4.0.11, if $g\colon \{+1,-1\}^k \to \{+1,-1\}$ is any hardness amplifying function, $\mathsf{EB}_{1-2\delta}(g) \geq 1 - \mathsf{NS}_{\frac{1}{2}-\delta}(g)$. But if we want to do hardness amplification within NP, g must be monotone, and so Kahn, Kalai, and

Linial's result Corollary 3.6.5 implies that $\mathsf{NS}_{\frac{1}{2}-\delta}(g) \leq \frac{1}{2} - \Omega(\frac{\log^2 k}{k})\delta$ and hence we have $\mathsf{EB}_{1-2\delta}(g) \geq \frac{1}{2} + \Omega(\frac{\log^2 k}{k})\delta$. Thus if we begin with a function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ with hardness at least $\frac{1}{2} + \Omega(\frac{\log^2 n}{n})$, then Theorem 4.0.9 will only tell us that $g \otimes f$ has hardness $\mathsf{EB}_{1-2\Omega(\frac{\log^2 n}{n})}(g) \geq \frac{1}{2} + \Omega(\frac{\log^2 k}{k})\Omega(\frac{\log^2 n}{n}) \geq \frac{1}{2} + \Omega(\frac{\log^2 (kn)}{kn})$. Thus as a function of the new input length kn, we are no harder after amplification than we were before. Therefore, hardness $\frac{1}{2} + \Omega(\frac{\log^2 m}{m})$ is a barrier for our techniques. Furthermore, it seems possible that $\mathsf{EB}_{1-2\delta}(g) \geq \frac{1}{2} + \Omega(k^{-\frac{1}{2}})$ for all monotone $g\colon \{+1,-1\}^k \to \{+1,-1\}$, which would make Theorem 4.1.1 even tighter for our techniques.

We close this section by showing how to remove the assumption that the initial hard function in NP is balanced, at the expense of a small loss in final hardness. We begin by employing a simple reduction using padding:

Proposition 4.5.3 *If there is a family of functions* (f_n) *in* NP *which is infinitely often* $(1 - 1/\mathrm{poly}(n))$*-hard for polynomial circuits, then for every* $\gamma > 0$ *there is a family of functions* (g_m) *in* NP *which is infinitely often* $(1 - 1/m^\gamma)$*-hard for polynomial circuits.*

Proof: Suppose (f_n) is infinitely often $(1 - 1/n^c)$-hard for polynomial circuits. Pick $K > c/\gamma$, and define $m = m(n) = n^C$. On an input x of length m, define $g_m(x)$ to be f_n 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 $(1 - 1/m^\gamma)$-hard for polynomial circuits. The proof is simple; suppose C were a circuit of size m^d that correctly computed g_m on a $1 - 1/m^\gamma$ fraction of the inputs in $\{+1,-1\}^m$. 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 $1 - 1/m^\gamma$ fraction of the remaining inputs, $\{+1,-1\}^n$. But now we have a circuit of size $n^{Kd} = \mathrm{poly}(n)$ which computes f_n with probability at least $1 - 1/n^{\gamma K} > 1 - 1/n^c$, a contradiction. □

We now outline the proof of Theorem 4.1.2. We begin with a $(1 - 1/n^\gamma)$-hard function f on n inputs by using Proposition 4.5.3, and we wish to apply Theorem 4.0.9 using the function g_k defined in Theorem 4.5.1. 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 $\pm L^{-1}$. At this point we can easily cook up a slightly altered version of f which is still hard and is within about $\pm L^{-1}$ of being balanced. This lets us apply Theorem 4.0.9, so long as $\eta/k \approx L^{-1}$. Since the expected bias we get from Corollary 4.5.2 exceeds $\frac{1}{2}$ by about $k^{-\frac{1}{2}}$, the largest η we would like to pick is $k^{-\frac{1}{2}}$, leading to $k^{-\frac{3}{2}} \approx L^{-1} \Rightarrow k^{-\frac{1}{2}} \approx L^{-\frac{1}{3}}$. This is why the exponent of $\frac{1}{3}$ arises in Theorem 4.1.2. We now give the rigorous proof:

Theorem 4.1.2 *If there is a function family (f_n) in NP which is infinitely often $(1 - 1/\mathrm{poly}(n))$-hard for circuits of polynomial size, then for any constant $\gamma > 0$ there is a function family (h_m) in NP which is infinitely often $(\frac{1}{2} + m^{-\frac{1}{3}+\gamma})$-hard for circuits of polynomial size.*

Proof: Let $\gamma > 0$ 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 4.5.3, there is a family of functions in NP, which we will also denote by (f_n), that is infinitely often $(1 - 1/n^\gamma)$-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 $h_m \equiv +1$.) Specifically, define $f'_n \colon \{+1,-1\}^{n+1} \to \{+1,-1\}$ as follows: on input yb, where $|y| = n$, $|b| = 1$, put

$$f'_n(yb) = \begin{cases} f(y) & \text{if } b = +1, \\ +1 & \text{if } b = -1,\ y \text{ is in the first } (i/8n^C)2^n \text{ strings of } \{+1,-1\}^n \text{ in lex order,} \\ -1 & \text{otherwise.} \end{cases}$$

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, $\mathrm{bias}(f'_n) \leq \frac{1}{2} + 1/8n^C$. Further, note that f'_n is easily seen to be $(1 - 1/2n^\gamma)$-hard for polynomial circuits.

Now we continue the definition of h_m. Pick $k = n^{\frac{2}{3}C}$, an integer, and let g_k be the function from Theorem 4.5.1 with parameter $\alpha = 2\gamma/C$. Note that with this choice, $1/2n^\gamma > 1/k^\alpha$ 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 $h_m(x) = g_k \otimes f'_n$, 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 $(\frac{1}{2}+m^{-\frac{1}{3}+O(1)\gamma})$-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, $\text{bias}(f'_n) \leq \frac{1}{2} + 1/8n^C$. For this m, we claim that h_m has the desired hardness. This follows in a straightforward manner from Theorem 4.0.9, using Corollary 4.5.2. To be exact, let $\epsilon = 1/2n^\alpha$, $\eta = 1/n^{C/3}$, and $\delta = 1$. Note that $\text{bias}(f'_n) \leq \frac{1}{2} + 1/8n^C \leq (1-2\epsilon)\eta/4k$, as needed. Theorem 4.0.9 tells us that h_m is $(\mathsf{EB}_\epsilon(g_k)+\eta)$-hard for polynomial circuits. Recall that $\epsilon > 1/k^\alpha$. Thus by Corollary 4.5.2, the hardness of h_m is at most $\frac{1}{2} + k^{-\frac{1}{2}+4\gamma/C} + 1/n^{C/3} = \frac{1}{2} + n^{-C/3+(8/3)\gamma} + n^{-C/3}$. As a function of the input length, $m \leq (n+2)^{C+1}$, the quantity $n^{-C/3+(8/3)\gamma} + n^{-C/3}$ can be made smaller than $m^{-\frac{1}{3}+O(1)\gamma}$ by taking $C = O(1/\gamma)$. □

4.6 A corresponding easiness theorem

In this section we demonstrate that Theorem 4.0.9 is close to being tight.

Definition 4.6.1 *We say a boolean function* $f\colon \{+1,-1\}^n \to \{+1,-1\}$ *is* $(1-\epsilon)$-easy for circuits of size s *if there is a circuit of size* s *which correctly computes* f *on a* $1-\epsilon$ *fraction of the inputs* $\{+1,-1\}^n$.

We shall show how to construct, for any ϵ and any g, a balanced function f which is $(1-\epsilon)$-hard, yet such that $g \otimes f$ is roughly $\mathsf{EB}_{2\epsilon}(g)$-easy. The constructions of the functions used in this section are essentially those of Shaltiel [Sha01].

Proposition 4.6.2 *There is a universal constant* $\gamma \geq .1$ *such that there exists a balanced function* $h\colon \{+1,-1\}^n \to \{+1,-1\}$ *which is* $(\frac{1}{2}+2^{-\gamma n})$*-hard for size* $2^{\gamma n}$.

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

Definition 4.6.3 *For a given rational number* $\epsilon \in [0,1]$*, define a canonical constant* z_ϵ *and a canonical constant-sized circuit* Z_ϵ *on* z_ϵ *inputs such that* $\mathbf{P}[Z_\epsilon = +1] = \epsilon$.

Definition 4.6.4 *Given a boolean function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ and a rational constant $\epsilon \in [0,1]$, define $f^{(\epsilon)}\colon \{+1,-1\}^{n+z_\epsilon+1}$ by*

$$f^{(\epsilon)}(x,a,b) = \begin{cases} f(x) & \textit{if } Z_\epsilon(a) = +1, \\ b & \textit{else.} \end{cases}$$

The following facts are easily verified:

Proposition 4.6.5

- *If f is balanced, so is $f^{(\epsilon)}$.*
- *If f is α-hard for size s, then $f^{(\epsilon)}$ is $(1-\epsilon+\epsilon\alpha)$-hard for size s.*
- *$f^{(\epsilon)}$ is $(1-\epsilon)$-easy for size $O(1)$.*

Definition 4.6.6 *If $g\colon \{+1,-1\}^n \to \{+1,-1\}$, 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 that, 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 $+1$ 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 4.0.9 allows for a tradeoff, depending on the comparative values of $\mathsf{SIZE}(g)$ and $\mathsf{R\text{-}B\text{-}SIZE}(g)$.

Theorem 4.6.7 *Let ϵ be a rational constant. Let $\epsilon' = \epsilon/(1-2^{-\gamma n}) = \epsilon + O(2^{-\gamma n})$, where γ is the constant from Proposition 4.6.2. Then there is a balanced function $f\colon \{+1,-1\}^{n+O(1)} \to \{+1,-1\}$ which is $(1-\epsilon)$-hard for circuits of size $2^{\gamma n}$, such that*

1. *$g \otimes f$ is $\mathsf{EB}_{2\epsilon'}(g)$-easy for size $\mathsf{R\text{-}B\text{-}SIZE}(g) + O(k)$;*
2. *$g \otimes f$ is $(1 - \mathsf{NS}_{\epsilon'}(g))$-easy for size $\mathsf{SIZE}(g) + O(k)$; and,*
3. *$g \otimes f$ is $(\mathsf{EB}_{2\epsilon'}(g) - O(m^{-\frac{1}{2}}))$-easy for size $m\mathsf{SIZE}(g) + O(mk)$.*

Proof: Take h to be the hard balanced function from Proposition 4.6.2, and let $f = h^{(2\epsilon')}$. Then by Proposition 4.6.5, f is balanced and $(1-\epsilon)$-hard for size $2^{\gamma n}$. But it is also $(1-2\epsilon')$-easy for size $O(1)$. We are now essentially in the Hard-Core scenario described in Section 4.2; with probability $(1-2\epsilon')$ we *know* the correct answer, and with probability $2\epsilon'$ we have no good idea.

Let us call our inputs $w_1, \dots, w_k$, where $w_i = (w_i', a_i, b_i)$. Let A be the constant-sized circuit which on input $(w_i', a_i, b_i) \in \{+1,-1\}^{n+z_{\epsilon'}+1}$ outputs b_i if $Z_{\epsilon'}(a_i) \neq +1$, or $\star$ if $Z_{\epsilon'}(a_i) = +1$. Let B be the circuit of size $O(k)$ which applies a copy of A to each input w_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 $R^k_{2\epsilon'}$.

Now we apply the two guessing strategies suggested in Section 4.2, 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. A standard 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 $a = 2(\mathrm{bias}(g_\rho) - \frac{1}{2})$, and assume without loss of generality that g_ρ is biased towards $+1$. 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 is straightforward to show using calculus that $a\exp(-\frac{a^2}{1+a}\frac{m}{4})$ is maximized for $a = \Theta(m^{-\frac{1}{2}})$. In this case, $a\eta = O(m^{-\frac{1}{2}})$. Averaging over ρ, we get the claimed result. □

Chapter 5

Learning Theory

In this chapter we give new computational learning algorithms whose efficiency and correctness are based on noise sensitivity considerations. Noise sensitivity was first related to computational learning theory by Bshouty, Jackson, and Tamon [BJT99b] in 1999. However, their results were mainly negative; they gave lower bounds for the amount of time necessary to learn classes of functions with high noise sensitivity under attribute noise. In contrast, our results are positive; we give new algorithms which work in the usual, noiseless model.

We first observe that any class of functions with low noise sensitivity can be learned under the uniform distribution in a correspondingly low amount of time. As a result, we can immediately apply our work from Sections 3.10, 3.11, and 3.12 to get new polynomial time and quasipolynomial time algorithms for learning various classes based on halfspaces, under the uniform distribution.

Next we describe a variation on the usual uniform-distribution PAC learning model, namely the Random Walk model, first studied by Bartlett, Fischer, and Höffgen [BFH94]. Briefly, in this model the learner's examples are not distributed independently, but rather are produced according to a random walk on the hypercube. Under the Random Walk model, we give a polynomial time algorithm for learning one of the most important and notorious classes in learning theory, namely polynomial-sized DNF formulas. This is the first known efficient algorithms for learning DNF in a natural, passive model of learning from random examples only. Our technique involves introducing another model of learning which we call the "Noise Sensitivity" model. We first show that learners with access to the

Random Walk model can simulate access to the Noise Sensitivity model. We then show how the relationship between noise sensitivity and Fourier coefficients allows the learner to identify large Fourier coefficient of the unknown DNF. This leads to an efficient learning algorithm in the Noise Sensitivity model by known learning theory techniques.

Finally, in our last result we work further on the problem of learning DNF under the uniform distribution by attacking the important problem of learning juntas. Although our techniques do not use noise sensitivity directly, we believe that breakthroughs for this problem may involve noise sensitivity. As Håstad and Bourgain showed (see Theorem 1.2.2), juntas and noise sensitivity are intimately related — sufficiently noise-stable functions must be close to juntas, and vice versa. Furthermore, the recent polynomial time algorithm of Fischer, Kindler, Ron, Safra, and Samorodnitsky [FKR$^+$02] for *testing* juntas is based on Fourier analysis of the sort relevant to noise sensitivity.

5.1 PAC learning preliminaries

The computational learning model we consider is Valiant's popular "Probably Approximately Correct" (PAC) model of learning from random examples [Val84]; see Kearns [Kea90] for an extensive survey. In this framework, a learning problem is identified with a *concept class* $\mathcal{C} = \cup_{n\geq 1} \mathcal{C}_n$ which is simply a collection of boolean functions, each $f \in \mathcal{C}_n$ being a function $\{+1,-1\}^n \to \{+1,-1\}$.

The goal of a learning algorithm A for $\mathcal{C}$ is to identify an unknown function $f \in \mathcal{C}$ by using random examples from this function only. In particular, the probabilistic algorithm A takes as input as *accuracy parameter* ϵ and a *confidence parameter* δ; it also has access to an *example oracle* $\mathsf{EX}(f, \mathcal{D})$. Here f may be any function in $\mathcal{C}_n$ and $\mathcal{D}$ may be any probability distribution over $\{+1,-1\}^n$. When queried, the example oracle provides the learning algorithm with a labeled *example* $\langle x, f(x)\rangle$, where x is drawn from the distribution $\mathcal{D}$. The output of A is a *hypothesis* h, which is a boolean function $h\colon \{+1,-1\}^n \to \{+1,-1\}$ (in the form of a circuit, say). The hypothesis h is said to be ϵ-*close* to f if $\Pr_{x\leftarrow\mathcal{D}}[h(x) = f(x)] \geq 1-\epsilon$. We say that A is a learning algorithm for $\mathcal{C}$ if for all $f \in \mathcal{C}$ and $\mathcal{D}$, when A is run with example oracle $\mathsf{EX}(f, \mathcal{D})$, with probability at least $1-\delta$ it outputs a hypothesis which is ϵ-close to f. Here the probability is over the random examples A sees from the oracle and also over A's own random bits.

The measure of A's efficiency is its running time; A is only charged unit time for each example it draws, but it is charged all of the time it takes to write down its hypothesis. In general, we consider A's running time as a function of n, ϵ^{-1}, $\log(1/\delta)$, and usually also a "size" parameter s from the concept class.

Finally, since PAC learning in its full generality seems to be very difficult for many natural concept classes, often some of its requirements are relaxed. When the algorithm need only work when the distribution $\mathcal{D}$ is the uniform distribution over $\{+1, -1\}^n$, it is said to be a *uniform-distribution* learning algorithm. Uniform-distribution PAC learning is very frequently studied; a very small selection of results includes [Ver90, HM91, LMN93, BFJ+94, Man95, Kha95, BT96, Jac97, BJT99a, Ser01, JKS02, KS03]. We will mostly be concerned with uniform-distribution PAC learning in this chapter. Another weakening of the model is to allow the learning algorithm to make *membership queries*; in this model, the learner is allowed to ask for the value of the target function on points of its choosing. This model gives the learner considerably more power than usual and is thus a significant weakening. It also departs from the traditional passive nature of learning from random examples. The new algorithms we present will never require membership queries; however, we shall often point out the cases in which allowing membership queries makes our learning problems easier.

5.1.1 Uniform-distribution learning via Fourier coefficients

Many uniform-distribution learning algorithms work by estimating the Fourier coefficients of the unknown function. This technique originated with the "Low Degree" algorithm of Linial, Mansour, and Nisan [LMN93]. (See Mansour's survey [Man94] for an overview and for proofs of some of the facts below.) Linial et al. showed that a learning algorithm with access to uniformly random examples from a function f could accurately estimate any particular Fourier coefficient $\hat{f}(S)$ it wanted in polynomial time. The technique is straightforward: since $\hat{f}(S) = \mathbf{E}[f(x)\chi_S(x)]$, drawing many uniformly random examples $\langle x_i, f(x_i)\rangle_{i=1\ldots m}$ and computing the average value of $f(x_i)\chi_S(x_i)$ gives a good estimate. To be exact,

Theorem 5.1.1 *Given access to uniform random examples from $f\colon \{+1,-1\}^n \to \{+1,-1\}$, for any $S \subseteq [n]$, a randomized algorithm can compute $\hat{f}(S)$ to within an additive error of λ with confidence $1 - \delta$ using $O(\lambda^{-2}\log(1/\delta))$ examples and time.*

Linial et al. also showed that for any boolean function f, if $h\colon \{+1,-1\}^n \to \mathbf{R}$ satisfies $\sum_{S\subseteq[n]}(\hat{f}(S)-\hat{h}(S))^2 \le \epsilon$ then the boolean function $\mathrm{sgn}(h)$ is ϵ-close to f. It follows quite easily that if an algorithm accurately estimates most of f's Fourier coefficients (all except for some whose squared weight sums to at most ϵ), then it can output a good hypothesis for f consisting of the sign of the truncated Fourier polynomial it computes. Rigorously, we have the following theorem (proved in [KM93]):

Theorem 5.1.2 *Suppose that a learning algorithm can determine a set of parities $\mathcal{S} \subseteq 2^{[n]}$ such that it is assured that*

$$\sum_{S\in\mathcal{S}} \hat{f}(S)^2 \ge 1-\epsilon.$$

Then with probability $1-\delta$ it can output an ϵ-close hypothesis for f in total time $\mathrm{poly}(|\mathcal{S}|, n, \epsilon^{-1}, \log(1/\delta))$.

We will use Theorem 5.1.2 as a black box. Since the quantity $\mathrm{poly}(|\mathcal{S}|, n, \epsilon^{-1}, \log(1/\delta))$ is invariably dominated by $|\mathcal{S}|$, we shall often say the running time is $\mathrm{poly}(|\mathcal{S}|)$, with the true quantity being understood.

5.2 Learning noise-stable functions

As we saw in Corollary 2.3.3 in Chapter 2, the relationship between the noise sensitivity of a function and its Fourier coefficients tells us that if a function has low noise sensitivity, then most of its Fourier coefficients are concentrated on low degree. Thus if we want to learn a concept class $\mathcal{C}$ under the uniform distribution, and we can show that the functions in $\mathcal{C}$ have bounded noise sensitivity, then we can get an efficient learning algorithm by taking $\mathcal{S}$ to be all the parities of low degree in Theorem 5.1.2. In particular, combining Corollary 2.3.3 and Theorem 5.1.2 we immediately have the following main theorem:

Theorem 5.2.1 *Suppose $\mathcal{C} = \cup_{n\ge1}\, \mathcal{C}_n$ is a concept class such that for every $f \in \mathcal{C}_n$, $f\colon \{+1,-1\}^n \to \{+1,-1\}$, we have $\mathsf{NS}_\epsilon(f) \le m(\epsilon)$, where $m\colon [0,\frac{1}{2}] \to [0,1]$ is a continuous, strictly increasing function. Then there is a uniform-distribution learning algorithm for $\mathcal{C}$ running in time $n^{O(k)}\mathrm{polylog}(\delta^{-1})$, where $k = \frac{1}{m^{-1}(\epsilon/2.32)}$.*

The "slogan" by which one can remember the running time is "n to the power of the reciprocal of the inverse [of the noise sensitivity bound]."

Although it is derived in a simple way from facts already known, Theorem 5.2.1 is a potentially powerful way of obtaining new uniform-distribution learning results. For example, we shall shortly see that it can be used to give efficient algorithms for the historically difficult problem of learning functions of halfspaces.

It is interesting to try to understand from an intuitive perspective why noise stability should imply fast learning. A simple attempt at justification might go as follows: If a function f is noise-stable, its value on a given point y should be well correlated with its values on points near to y in Hamming distance. Thus a good way to learn f is to draw a number of labeled examples from the oracle, thus producing a "net" of points on which f's value is known; then the hypothesis can guess the value of f on a given point y by looking at the known values of f near y and computing an appropriate weighted threshold. The weight given to a known example should depend only on its Hamming distance from y.

Further reflection might lead one to view the above intuition with suspicion. It might seem that it should not be as easy as taking a net after all, since unless an exponential number of examples are drawn for the net, almost every new point to be hypothesized about will be at distance almost $\frac{n}{2}$ from the known net points. Knowing that f has low noise sensitivity at some small ϵ seems unhelpful in making such long-distance correlations.

However, the original intuition is correct, albeit not sophisticated enough to be a proof. To see this, simply open the learning black box employed in Theorem 5.2.1. The learning algorithm draws a number of examples, $\langle x_i, f(x_i)\rangle_{i=1\ldots m}$. The final hypothesis applied to a point y is given by

$$\mathrm{sgn}\left(\sum_{|S|\leq k} \tilde{\hat{f}}(S) y_S\right),$$

where $\tilde{\hat{f}}(S)$ is an estimate for $\hat{f}(S)$. Each of these estimates is formed by a simple average, $\tilde{\hat{f}}(S) = \frac{1}{m}\sum_{i=1}^{m} f(x_i)(x_i)_S$. Thus the hypothesis on y is

$$\begin{aligned}\mathrm{sgn}\left(\sum_{|S|\leq k} \frac{1}{m}\sum_{i=1}^{m} f(x_i)(x_i)_S y_S\right) &= \mathrm{sgn}\left(\sum_{i=1}^{m}\left(\sum_{|S|\leq k}(x_i y)_S\right) f(x_i)\right)\\ &= \mathrm{sgn}\left(\sum_{i=1}^{m} W_k(\Delta(y, x_i)) f(x_i)\right),\end{aligned}$$

where $W_k(d) = \sum_{s=0}^{k} K_s(d)$ and $K_s = K_s^{(n)}$ is the sth Kravchuk polynomial, $K_s(d) = \sum_{j=0}^{s}(-1)^j \binom{d}{j}\binom{n-d}{s-j}$. (For a brief reference on Kravchuk polynomials and sums of Fourier characters, see Linial and Samorodnitsky [LS03].) So the hypothesis we use for noise-stable f's is indeed a weighted threshold over the labels of the points in a net, with the weight given to a known example depending only on its Hamming distance from the point whose label we are guessing.

Still, the weighting scheme used is fairly strange; further, there does not seem to be an easy, direct way to show that it leads to a good hypothesis for noise-stable functions. Rather, it appears that any justification must pass to the Fourier representation. Note that a more natural-seeming weighting scheme for functions with low noise sensitivity at ϵ, namely $W(d) = \epsilon^d(1-\epsilon)^{n-d}$, appears to require far too many points in the net to work.

5.3 Learning intersections and other functions of halfspaces

In the context of learning theory, a linear threshold function $f\colon \mathbf{R}^n \to \{+1,-1\}$ given by $f(x) = \mathrm{sgn}(\sum_{i=1}^{n} w_i x_i - \theta)$ is usually referred to as a halfspace. The problem of learning an unknown halfspace from labeled data is one of the oldest problems in machine learning, dating back to the late 1950s [Ros58, Blo62]. This problem has been intensively studied over the years and efficient algorithms are now known for several different learning models. In particular, the concept class of halfspaces over $\mathbf{R}^n$ is PAC-learnable in polynomial time under any probability distribution: Blumer, Ehrenfeucht, Haussler, and Warmuth [BEHW89] showed that linear programming can be used to give a $\mathrm{poly}(n)\epsilon^{-1}\log(1/\delta)$ algorithm.

While the problem of learning a single halfspace is fairly well understood, learning more complicated functions which depend on several halfspaces seems to be quite difficult; in particular, learning an intersection of several unknown halfspaces stands as a major open problem in computational learning theory. Intersections of halfspaces form an important concept class for many reasons: any convex body can be expressed as an intersection of halfspaces, and several well-studied classes of boolean functions such as DNF formulas can be naturally viewed as special cases of intersections of halfspaces over $\{+1,-1\}^n$.

Given the apparent difficulty of learning intersections of halfspaces, several algorithms in weaker models have been proposed. Building on work of Blum, Chalasani, Goldman, and Slonim [BCGS98] and Baum [Bau91a], Kwek and Pitt [KP98] gave a membership query algorithm for learning the intersection of k halfspaces in $\mathbf{R}^n$ with respect to any probability distribution in time polynomial in n and k. Progress has been much more limited for learning intersections of halfspaces from random examples only; all such results to date require that the examples be drawn from some restricted class of probability distributions. Baum [Bau91b] gave a polynomial time algorithm for learning an intersection of exactly two zero-threshold halfspaces under any origin-symmetric distribution (i.e., one satisfying $\mathcal{D}(x) = \mathcal{D}(-x)$ for all $x \in \mathbf{R}^n$). His algorithm is essentially a reduction to the problem of learning a single halfspace, and does not seem to generalize to more than two halfspaces. Building on work of Blum and Kannan [BK97a], Vempala [Vem97] gave a polynomial time algorithm which can learn an intersection of $\log n / \log \log n$ halfspaces under "near-uniform" distributions on the Euclidean ball in $\mathbf{R}^n$.

Computational learning theory is frequently concerned with *boolean* halfspaces — i.e., restrictions of halfspaces to $\{+1, -1\}^n$ — and neither of the two algorithms mentioned previously can learn even the intersection of two arbitrary boolean halfspaces under the uniform distribution on $\{+1, -1\}^n$ in subexponential time. (Baum's algorithm comes close, but requires the halfspaces to have a threshold of zero; also, it does not work at all for the intersection of three boolean halfspaces.) We now give new polynomial and quasipolynomial time algorithms for this important problem of learning functions of boolean halfspaces under the uniform distribution. These are immediately derived from Theorem 5.2.1 and our noise sensitivity calculations in Sections 3.10, 3.11, and 3.12.

From Corollary 3.10.3 and Proposition 2.2.8 we get that any function of k boolean halfspaces has noise sensitivity at most $\frac{5}{4}k\sqrt{\epsilon}$ at ϵ. Thus by Theorem 5.2.1 we get the following:

Theorem 5.3.1 *Let $\mathcal{C}_n$ be the concept class of all boolean functions on $\{+1, -1\}^n$ expressible as some function of k boolean halfspaces. Then $\mathcal{C}_n$ can be learned under the uniform distribution to accuracy ϵ in time $n^{O(k^2/\epsilon^2)}$, assuming $\epsilon \leq 1/k^2$.*

Let us emphasize that for constant ϵ and k, this gives a polynomial time algorithm for learning any function of k halfspaces under the uniform distribution.

From Theorem 3.11.1 we derive a quasipolynomial time algorithm for learning the read-once intersection of halfspaces:

Theorem 5.3.2 *Let $\mathcal{C}_n$ be the concept class of all functions of the form* $\mathsf{AND}(h_1, \ldots, h_k)$*,* *where the h_i's are boolean halfspaces on disjoint subsets of n variables. Then $\mathcal{C}_n$ can be learned under the uniform distribution to accuracy ϵ in time $n^{O((\log k)/\epsilon^2)}$, assuming $\epsilon \leq 1/\log k$.*

In passing we note that Golea, Hancock, and Marchand [GHM94] gave a polynomial time algorithm for learning the read-once intersection of *majorities* under the uniform distribution. But the concept class mentioned in Theorem 3.11.1 is much richer as it allows for the intersection of arbitrary halfspaces.

Finally, from Theorem 3.12.1 we get a quasipolynomial time algorithm for learning the read-once majority (or unweighted threshold) of halfspaces:

Theorem 5.3.3 *Let $\mathcal{C}_n$ be the concept class of all functions of the form $\mathrm{sgn}(h_1 + \cdots + h_k - \theta)$, where the h_i's are boolean halfspaces on disjoint subsets of n variables. Then $\mathcal{C}_n$ can be learned under the uniform distribution to accuracy ϵ in time $n^{\tilde{O}(\log(k/\epsilon)/\epsilon^4)}$, assuming $\epsilon \leq 1/\log k$.*

We close by noting that Bourgain's Theorem 1.2.2 shows that Theorem 5.2.1 cannot be used to produce uniform-distribution learning algorithms which run faster than $n^{O(1/\epsilon^2)}$, except when they are for uninteresting concept classes — ones consisting of functions that are close to $O(1)$-juntas. Thus Theorem 5.3.1, with k constant, is an example of the fastest nontrivial learning algorithm derivable from Theorem 5.2.1.

5.4 Learning DNF from random walks

Disjunctive normal form (DNF) formulas of polynomial size form what is probably the most notorious concept class studied in computational learning theory. DNF formulas seem to be a natural form of knowledge representation for humans; furthermore, they are a universally expressive concept class in the sense that every boolean function can be represented as a DNF formula of some size. In his original paper on PAC learning, Valiant [Val84] asked whether there is a polynomial time learning algorithm for the class of DNF formulas of polynomial size. Such algorithms have proved extremely difficult to come by; to date, the fastest known algorithm runs in time $2^{\tilde{O}(n^{1/3})}$ [KS01]. Even under the uniform distribution the fastest known learning algorithm [Ver90] requires superpolynomial time $n^{O(\log(n/\epsilon))}$. A breakthrough on this problem was made by Jackson [Jac97], who gave a polynomial time membership query algorithm for learning polynomial-sized DNF under the uniform distribution. His algorithm combined a membership query algorithm of Kushilevitz and Mansour [KM93] for finding Fourier coefficients with the learning-theoretic concept of boosting.

Unfortunately, granting the learner membership queries departs significantly from the traditional passive model of learning from random examples only. Building on Jackson's and Kushilevitz's and Mansour's work, Bshouty and Feldman [BF02] gave a polynomial time learning algorithm polynomial-sized DNF under a model of intermediate power between uniform-distribution learning and uniform-distribution learning with membership queries. In their model, called SQ-$\mathcal{D}_\rho$, the learner is allowed to make statistical queries about the target function under product distributions of the learner's choosing. Although this gives the learner strictly less power than a learner with membership queries, it still represents a non-passive (and somewhat artificial) model of learning.

In this section we will consider a natural, passive model of learning under the uniform distribution, the Random Walk model. This model is also of intermediate power compared to uniform-distribution learning and uniform-distribution learning with membership queries. We will give a polynomial time algorithm for learning polynomial-sized DNF in the Random Walk model. We emphasize that this is the first known polynomial time algorithm for learning a universally expressive concept class in a passive model of learning from random examples only.

5.4.1 The Random Walk learning model

Variants of PAC learning in which the examples are not i.i.d., but rather are generated according to a stochastic process, were first considered by Aldous and Vazirani [AV90]. Despite being quite natural, these models have not been studied nearly as intensively as other variants on PAC learning. Bartlett, Fischer, and Höffgen [BFH94] introduced what is perhaps the simplest and most natural such model, namely the Random Walk model, which we define shortly. The authors gave learning algorithms for some very simple concept classes under the Random Walk model, namely boolean threshold functions in which each weight is 0 or 1, parities of two monotone ANDs, and DNF formulas with two terms. Gamarnik [Gam99] further studied learning under stochastic processes but did not give any algorithms for specific concept classes.

Let us begin by defining the Random Walk model of learning. As the Random Walk model is a variation on the standard PAC model, we shall only describe the differences. In the Random Walk model our concept classes $\mathcal{C}_n$ will consist of collections of *real-valued* boolean functions $f\colon \{+1,-1\}^n \to \mathbf{R}$. A learning algorithm's hypothesis is also required to be a real-valued boolean function h, and the measure of its error with respect to the target function f is $\mathbf{E}[(f-h)^2]$. This extension to real-valued boolean functions is very common in PAC learning. The major difference between the two models is the way in which the learner gets its examples. The first labeled example the learner gets is a random point from $\{+1,-1\}^n$. Following this, the examples the learner sees are generated by a standard random walk on the hypercube. That is, if the tth example given to the learner is x, then the $(t+1)$st example will be chosen by selecting a bit position $i \in [n]$ uniformly at random and then flipping the ith bit of x.

Let us compare the Random Walk model with other common PAC learning models. First, we note that learning real-valued boolean functions under the L_2 error measure is a strict generalization of learning boolean-valued boolean functions under the usual uniform-distribution error metric. This is because, as noted in Subsection 5.1.1, taking the sign of a real-valued function which has L_2 distance ϵ from a boolean-valued function yields a boolean-valued function with ϵ error under the uniform distribution.

Second, we note that learning boolean-valued functions under the Random Walk model is no stronger than learning under the usual uniform distribution model. The reason is that a learner with access to examples generated by a Random Walk can easily simulate examples generated from the uniform distribution with only polynomial slowdown. This is because the random walk on the hypercube mixes rapidly; if a learner getting examples from the Random Walk lets $O(n \log n)$ steps pass then the next example it gets will be uniformly random.[1]

Next, it is immediate that having access to membership queries is at least as powerful as getting examples generated from a random walk. In fact, we now give a proof that uniform-distribution learning with membership queries is strictly easier than learning in the Random Walk model, under a standard cryptographic assumption.

Proposition 5.4.1 *If one-way functions exist then there is a concept class $\mathcal{C}$ which is*

[1] Strictly speaking, the example will only be very nearly uniformly random; we should factor the confidence error δ into this statement. However, in this section all considerations involving δ are completely standard and we will frequently gloss over them for clarity.

learnable in polynomial time under the uniform distribution with membership queries, but is not learnable in polynomial time in the Random Walk model.

Proof: Assume one-way functions exist; then pseudorandom function families also exist by a well known result in cryptography [HILL99]. Let $\{f_s \colon \{+1,-1\}^n \to \{+1,-1\}\}_{s\in\{+1,-1\}^n}$ be a pseudorandom function family. For $s \in \{+1,-1\}^n$ let $g_s \colon \{+1,-1\}^n \to \{+1,-1\}$ be defined by

$$c_s(x) = \begin{cases} s_i & \text{if } x = e_i \text{ for some } i \in [n], \\ f_s(x) & \text{otherwise.} \end{cases}$$

(Here e_i denotes the string $(-1,\ldots,-1,+1,-1,\ldots,-1)$, with the $+1$ in the ith position.) We claim the concept class $\mathcal{C} = \{g_s\}_{s\in\{+1,-1\}^n}$ has the desired property.

It is easy to see that any $g_s \in \mathcal{C}$ can be learned exactly in polynomial time if membership queries are allowed. The algorithm simply queries $e_1,\ldots,e_n$ to learn all bits $s_1,\ldots,s_n$ of s and outputs a representation of g_s. On the other hand, a random walk which proceeds for only poly(n) steps will with probability $1-2^{-\Omega(n)}$ miss all the points e_i. A straightforward argument shows that conditioned on missing all these points, it is impossible to learn g_s in polynomial time. (To see this, note that an algorithm which has oracle access to a pseudorandom function f_s can easily simulate a random walk which misses all e_i. Thus if it were possible to learn g_s in polynomial time from a random walk conditioned on missing all e_i, it would be possible to learn the class $\{f_s\}$ given oracle access to f_s. But this is easily seen to contradict the definition of a pseudorandom function family.) $\Box$

We end this subsection by describing an equivalent model to the Random Walk model which is easier to work with for our purposes. We call this model the Random Walk model with *updating oracle*. The updating oracles gives examples to the learner in a slightly different fashion than does the usual Random Walk oracle. In the updating oracle, again the first example given is chosen uniformly random. Further, the example given at time $t+1$ depends on the example from time t. Suppose this previous example was x. Then for the $(t+1)$st example, the updating oracle picks an index i uniformly at random. The oracle then *updates* the ith bit of x, producing y. That is, y is equally likely to be x or $\sigma_i x$. Finally, the updating oracle tells the learner $\langle i, y, f(y)\rangle$.

It is easy to simulate the updating oracle using the original Random Walk oracle. Each time the learner wants a new example, it tosses a fair coin. On heads, it draws a new example from the standard Random Walk oracle, noting which input bit got flipped. On tails, it chooses a random bit position i and pretends that the updating oracle announced that the ith bit was updated but did not change.

5.4.2 A Noise Sensitivity learning model

As we have seen, learning under the Random Walk model is easier than learning under the uniform distribution. In this section we introduce a new family of learning models of strength intermediate to these two. We call these models the "Noise Sensitivity models" of learning. Although the models are somewhat artificial, they are easy to define and are the least powerful passive models we know of in which polynomial-sized DNF can be learned efficiently. Since we will see that the Random Walk model can simulate the Noise Sensitivity models, we conclude that DNF can also be learned efficiently in the Random Walk model.

For each value of $\gamma \in [0, \frac{1}{2}]$, we define the γ-Noise Sensitivity model of learning as follows: The model is again a model for learning real-valued boolean functions under the uniform distribution. The distinguishing feature is the nature of the example oracle. Given an unknown target function $f\colon \{+1,-1\}^n \to \mathbf{R}$, the learner has access to the "Noise Sensitivity oracle," $\mathsf{NS\text{-}EX}_\gamma(f)$. Every time the learner asks for an example, $\mathsf{NS\text{-}EX}_\gamma(f)$ independently chooses a random input $x \in \{+1,-1\}^n$, forms $y = N_\gamma(x)$, and then tells the learner $\langle x, f(x), y, f(y)\rangle$. Note that this oracle is of equivalent power to an "updating" Noise Sensitivity oracle, in which each bit of x is updated with probability 2γ, and the learner is told which bits were updated in going from x to y. To see this, simply note that the extra information can be simulated by the learner with access to the usual $\mathsf{NS\text{-}EX}_\gamma(f)$ oracle: upon seeing a pair of examples (x, y), the learner decides that each bit position in which x and y differ was updated; furthermore, for each bit position on which x and y are the same, the learner pretends that an update occurred independently with probability $\gamma/(1-\gamma)$.

Let us consider the different models of learning we get as we vary γ. The cases $\gamma = 0$ and $\gamma = \frac{1}{2}$ are trivially equivalent to the usual PAC model of learning under the uniform distribution. For values $\gamma \in (0, \frac{1}{2})$, learning with $\mathsf{NS\text{-}EX}_\gamma(f)$ is clearly at least as easy as learning under the uniform distribution. Since we shall show that polynomial-sized DNF are efficiently learnable in in the γ-Noise Sensitivity model for every constant $\gamma \in (0, \frac{1}{2})$, it appears as though learning in these models is *strictly* easier than learning under the uniform distribution. It also seems to us that for differing constants $\gamma, \gamma' \in (0, \frac{1}{2})$ the γ- and γ'-Noise

Sensitivity models are of incomparable strength.

One thing we can show explicitly is that having access to the Random Walk oracle is at least as powerful as having access to $\mathsf{NS\text{-}EX}_\gamma$ for any γ:

Proposition 5.4.2 *For any $\gamma \in [0, \frac{1}{2}]$, any γ-Noise Sensitivity learning algorithm can be simulated in the Random Walk model with only a multiplicative $O(n \log n)$ slowdown in running time.*

Proof: Fix $\gamma \in [0, \frac{1}{2}]$. We show how to simulate the oracle $\mathsf{NS\text{-}EX}_\gamma$ using the Random Walk model's updating oracle. To get an example $\langle x, f(x), y, f(y)\rangle$, we first draw $O(n \log n)$ examples from the updating oracle to get to a uniformly random point x; this point and its label $f(x)$ will be the first part of our $\mathsf{NS\text{-}EX}_\gamma$ example. We then want to generate a point y which is formed from x by *updating* each bit with probability 2γ. This is equivalent to picking a quantity $u \sim \mathrm{Bin}(n, 2\gamma)$, and then updating a random subset of u of x's bits. Accordingly, in our simulation we shall randomly choose an integer $0 \leq u \leq n$ according to $\mathrm{Bin}(n, 2\gamma)$. We then repeatedly draw examples from the Random Walk updating oracle until u distinct input positions have been updated. Having done this, it as if a random subset of u bits had been updated, since updating an input position more than once has no extra effect. Therefore, if we call the resulting point y and output $\langle x, f(x), y, f(y)\rangle$, then this example is distributed exactly as it should be for the oracle $\mathsf{NS\text{-}EX}_\gamma$. Note that even if u is as large as n, it only takes $O(n \log n)$ samples to get a string in which all $u = n$ distinct bit positions of x have been updated. □

Our main theorem in this section is the following:

Theorem 5.4.3 *The class of DNF formulas on n variables with s terms can be learned in the Random Walk model in time* $\mathrm{poly}(n, s, \epsilon^{-1}, \log(1/\delta))$.

5.4.3 Performing the Bounded Sieve in the Noise Sensitivity model

As stated earlier, we prove Theorem 5.4.3 by showing that polynomial-sized DNF can be learned under any γ-Noise Sensitivity learning model. Then Theorem 5.4.3 follows immediately from Proposition 5.4.2. We therefore prove the following:

Theorem 5.4.4 *Let $\gamma \in (0, \frac{1}{2})$, and let $c_0 = -\log(\gamma(\frac{1}{2} - \gamma))$, a constant if γ is constant. Then the class of polynomial-sized DNF formulas on n variables can be learned in the γ-Noise Sensitivity model in time* $\mathrm{poly}(n^{c_0}, \epsilon^{-c_0}, \log(1/\delta))$.

To prove Theorem 5.4.4, we give an algorithm that, given access to the oracle $\mathsf{NS\text{-}EX}_\gamma(f)$, finds all of the "large" Fourier coefficients $\hat{f}(S)$ of f which satisfy $|S| \leq O(\log n)$. More precisely, we prove the following:

Theorem 5.4.5 *Let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be an unknown function, and let $\gamma \in (0,\frac{1}{2})$. Fix parameters $\ell \in [n]$ and $\theta > 0$. Then there is an algorithm running in time $\mathrm{poly}(n, [\gamma(\frac{1}{2}-\gamma)]^{-\ell}, \theta^{-1}, ||f||_\infty, \log(1/\delta))$ which, given access to examples from the oracle $\mathsf{NS\text{-}EX}_\gamma(f)$, with probability $1-\delta$ returns a list of subsets of $[n]$ such that*

- *for each $S \subseteq [n]$, if $|S| \leq b$ and $\hat{f}(S)^2 \geq \theta$, then S is in the list; and,*
- *for each set S in the list, $|S| \leq b$ and $\hat{f}(S)^2 \geq \theta/2$.*

(Here $||f||_\infty$ denotes $\max_{x\in\{+1,-1\}^n} |f(x)|$.)

Bshouty and Feldman call the task performed by this algorithm the *Bounded Sieve.* It is a weakened version of the algorithm of Kushilevitz and Mansour which finds *all* large Fourier coefficients $\hat{f}(S)$, regardless of $|S|$. As noted in Bshouty and Feldman's paper [BF02], Jackson's Harmonic Sieve algorithm for learning DNF does not need the full power of Kushilevitz and Mansour's routine; it only requires that it work for sets S whose size is bounded by $O(\log n)$ (and that it works in time polynomial in $||f||_\infty$). Our Theorem 5.4.5 is the analogue of Bshouty and Feldman's Theorem 13, and the fact that Theorem 5.4.4 follows from Theorem 5.4.5 follows by repeating the arguments following Theorem 13 in [BF02]. Thus to complete the proof of Theorem 5.4.3, we now prove Theorem 5.4.5.

5.4.4 Proof of Theorem 5.4.5

The idea for proving Theorem 5.4.5 is to get at the Fourier coefficients of the unknown function indirectly via a noise sensitivity-like quantity:

Definition 5.4.6 *Given $f\colon \{+1,-1\}^n \to \mathbf{R}$, $\gamma \in (0,\frac{1}{2})$, and $I \subseteq [n]$, define*

$$\mathcal{T}_\gamma^{(I)}(f) = \sum_{S \supseteq I} (1-2\gamma)^{|S|} \hat{f}(S)^2.$$

When f and γ are clear from context, we write simply $\mathcal{T}(I)$.

(Notice that $\mathcal{T}_\gamma^{(\emptyset)}(f) = 1 - 2\mathsf{NS}_\gamma(f)$.)

We can use the Noise Sensitivity oracle to estimate the quantities $\mathcal{T}_\gamma^{(I)}(f)$:

Lemma 5.4.7 *For fixed constant* $\gamma \in (0, \frac{1}{2})$ *and unknown* $f \colon \{+1, -1\}^n \to \mathbf{R}$*, an algorithm with access to* $\mathsf{NS\text{-}EX}_\gamma(f)$ *can, with probability* $1 - \delta$*, estimate* $\mathcal{T}(I)$ *to within* $\pm\eta$ *in time* $\mathrm{poly}(n, \gamma^{-|I|}, ||f||_\infty, \eta^{-1}, \log(1/\delta))$.

Proof: Given γ and I, consider the joint probability distribution $\mathcal{D}_\gamma^{(I)}$ defined over pairs of strings $(x, y) \in (\{+1, -1\}^n)^2$ as follows: First x is picked uniformly at random; then y is formed by updating each bit of x in I with probability 1 and updating each bit of x not in I with probability 2γ. We claim that access to pairs from this distribution and their values under f can be simulated by access to $\mathsf{NS\text{-}EX}_\gamma(f)$, with slowdown $\mathrm{poly}(\gamma^{-|I|})$. To see this, simply use the "updating" version of the $\mathsf{NS\text{-}EX}_\gamma(f)$ oracle and reject all samples in which not every bit in I is updated. Conditioning on not rejecting we indeed get pairs precisely from the distribution $\mathcal{D}_\gamma^{(I)}$; furthermore, the time it takes to get a good sample is $\mathrm{poly}(\gamma^{-|I|})$ with high probability.

Let us define $\mathcal{T}'(I)$ to be $\mathbf{E}_{(x,y) \leftarrow \mathcal{D}_\gamma^{(I)}}[f(x)f(y)]$. Since we can simulate access to pairs from $\mathcal{D}_\gamma^{(I)}$ and their values under f, we can estimate $\mathcal{T}'(I)$ simply by taking many samples and averaging. By standard arguments we can compute a $\pm\eta$ approximation with probability $1 - \delta$ in time $\mathrm{poly}(n, ||f||_\infty, \eta^{-1}, \log(1/\delta))$. But the quantity $\mathcal{T}'(I)$ is very closely related to $\mathcal{T}(I)$. In particular, an easy argument akin to the proof of Proposition 2.3.1 shows that $\mathcal{T}'(I) = \sum_{S \colon S \cap I = \emptyset}(1 - 2\gamma)^{|S|}\hat{f}(S)^2$. Let us now define $\mathcal{T}''(I) = \mathcal{T}'(\emptyset) - \mathcal{T}'(I)$, again a quantity we can estimate in time $\mathrm{poly}(n, \gamma^{-|I|}, ||f||_\infty, \eta^{-1}, \log(1/\delta))$. We have $\mathcal{T}''(I) = \sum_{S \colon S \cap I \neq \emptyset}(1 - 2\gamma)^{|S|}\hat{f}(S)^2$. Thus if we compute $\mathcal{T}''(J)$ for all $J \subseteq I$, it is straightforward to calculate $\mathcal{T}(I) = \sum_{S \supseteq I}(1 - 2\gamma)^{|S|}\hat{f}(S)^2$ using inclusion-exclusion. Since there are only $2^{|I|} \leq \gamma^{-|I|}$ such subsets J, the claimed running time follows. □

Next, we note that the sum of the $\mathcal{T}(I)$ values across all $|I| = j$ is not too large:

Lemma 5.4.8 *For any* $f \colon \{+1, -1\}^n \to \mathbf{R}$ *and* $\gamma \in (0, \frac{1}{2})$*, we have*

$$\sum_{|I| = j} \mathcal{T}(I) \leq ||f||_\infty^2 (2\gamma)^{-j}.$$

Proof: We have

$$\begin{aligned}\sum_{|I|=j} \mathcal{T}(I) &= \sum_{|I|=j} \sum_{S \supseteq I} (1-2\gamma)^{|S|} \hat{f}(S)^2 \\ &= \sum_{|S| \geq j} \binom{|S|}{j} (1-2\gamma)^{|S|} \hat{f}(S)^2 \\ &\leq \sum_{|S| \geq j} \hat{f}(S)^2 \sum_{t=j}^{\infty} \binom{t}{j} (1-2\gamma)^t \\ &= ||f||_2^2 \frac{1}{1-2\gamma} \left(\frac{1-2\gamma}{2\gamma}\right)^{j+1} \\ &\leq ||f||_\infty^2 (2\gamma)^{-j},\end{aligned}$$

as claimed. □

Using these lemmas we can now complete the proof of Theorem 5.4.5. Consider the directed graph on all subsets of $[n]$, in which there is an edge from I to J if $I \subset J$ and $|J \setminus I| = 1$. Imagine the nodes I being divided into n layers, according to the value of $|I|$. Our algorithm for finding the large Fourier coefficients of f will perform a breadth-first search on this graph, starting at the node $I = \emptyset$. For each active node in the search, the algorithm estimates both $\mathcal{T}(I)$ and $\hat{f}(I)^2$. Naturally, if the estimate of $\hat{f}(I)^2$ is at least $\theta/2$ then the algorithm adds I to the list of f's large Fourier coefficients. The breadth-first search proceeds to the neighbors of I only if $|I| < \ell$ and the estimate of $\mathcal{T}(I)$ is at least $(1-2\gamma)^\ell \theta/2$. We make two claims: first, the algorithm finds (with high probability) all Fourier coefficients $\hat{f}(S)$ with $\hat{f}(S)^2 \geq \theta$ and $|S| \leq \ell$; and second, the algorithm ends its search within time $\mathrm{poly}(n, [\gamma(\frac{1}{2} - \gamma)]^{-\ell}, \theta^{-1}, ||f||_\infty, \log(1/\delta))$.

To see the first claim, simply note that if $|S| \leq \ell$ and $\hat{f}(S)^2 \geq \theta$, then this Fourier coefficient contributes at least $(1-2\gamma)^\ell \theta$ to the value of $\mathcal{T}(I)$ for all $I \subseteq S$. Thus by the monotonicity of $\mathcal{T}$, the search will certainly proceed all the way to S, so long as all estimations are taken to be sufficiently precise (compared to the quantity $(1-2\gamma)^\ell \theta/2$).

For the second claim, note that by Lemma 5.4.8, the number of "active nodes" at layer j in the breadth-first search can be at most

$$\frac{||f||_\infty^2 (2\gamma)^{-j}}{(1-2\gamma)^j \theta/2} = 2||f||_\infty^2 \theta^{-1} \left(2\gamma(1-2\gamma)\right)^{-j}.$$

Since j is never more than ℓ, the total number of nodes the breadth-first search ever encounters is at most $2||f||_\infty^2 \theta^{-1} (2\gamma(1-2\gamma))^{-(\ell+1)} = \mathrm{poly}(||f||_\infty, \theta^{-1}, [\gamma(\frac{1}{2} - \gamma)]^{-\ell})$. Thus our estimations need only have additive error inverse-polynomial in this quantity, and we conclude that the final running time is indeed $\mathrm{poly}(n, [\gamma(\frac{1}{2} - \gamma)]^{-\ell}, \theta^{-1}, ||f||_\infty, \log(1/\delta))$, as claimed.

5.5 Learning juntas

In this final section of the chapter, we study the problem of learning juntas under the uniform distribution. We believe this is the most important open question in uniform-distribution PAC learning. As discussed in the previous section, polynomial-sized DNF formulas, along with polynomial-sized decision trees, are two very natural, important, and notorious classes that we would like to efficiently learn under the uniform distribution. However a basic bottleneck for learning these classes is that any $O(\log n)$-junta on n bits can be represented by a DNF formula or decision tree of polynomial size. Thus in order to make progress on learning DNF and decision trees, it is necessary to find algorithms that can learn juntas of size $O(\log n)$ or smaller. But furthermore, the reverse relation holds: any size-k decision tree is also a k-junta, and any k-term DNF is ϵ-indistinguishable (under the uniform distribution) from a $k\log(k/\epsilon)$-junta. Thus we conclude that the essential open problems of learning $\omega(1)$-sized decision trees and $\omega(1)$-term DNF in polynomial time are equivalent to the problem of learning $\omega(1)$-juntas in polynomial time. Along with the natural elegance of the problem, this justifies our assertion regarding the importance of the problem of learning juntas under the uniform distribution.

Let us now formally define the problem. We wish to learn the concept class of k-juntas over n variables. We shall think of the parameter k as being very small compared to n; in particular, we view it as $O(\log n)$ and possibly even as small as a large constant. We learn in the uniform-distribution PAC framework with the following simplification: the accuracy parameter ϵ will always taken to be 0; i.e., our hypotheses will be required to be exactly correct. This assumption does not lose much generality since any hypothesis which is ϵ-close to a k-junta for $\epsilon < 2^{-k}$ must have zero error. Since running time dependence should be polynomial in ϵ^{-1}, fixing ϵ to be, say, 2^{-k-1} incurs a running time blowup of only $\mathrm{poly}(2^k)$. When measuring running time dependencies we view factors of $\mathrm{poly}(2^k)$ as negligible, since for $k = O(\log n)$ they are polynomial. We shall therefore typically express running times in the form $\mathrm{poly}(n, 2^k, \log(1/\delta)) \cdot n^\alpha$ and view n^α as the essential measure of complexity.

The naive algorithm for learning k-juntas does a brute-force search over all possible $\binom{n}{k}$

sets of relevant variables. For $k = O(\log n)$ there are at least $n^{k-\text{polylog}(k)}$ such sets, and thus the naive algorithm takes time at least $n^{(1-o(1))k}$. In this section we make the first known significant improvement to this running time, giving an algorithm which learns k-juntas under the uniform distribution in time $\text{poly}(n, 2^k, \log(1/\delta) \cdot n^{\frac{\omega}{\omega+1}k}$, where ω is the matrix multiplication constant. Since Coppersmith and Winograd [CW90] showed $\omega < 2.376$, our running time is about $n^{.704k}$.

5.5.1 History of the problem

The problem of learning juntas is a natural one in the context of machine learning. One of the most important and challenging issues in machine learning is how to learn efficiently and effectively in the presence of irrelevant information. In real-world learning scenarios, a frequently encountered situation is one in which the data points one sees contain a large amount of information, but the labeling on them one is trying to learn depends only on a small unknown portion of this information. For example, in a computational biology scenario each data point may correspond to a long DNA sequence and the label may be some property which depends only on a small unknown active part of this sequence. To model this phenomenon, Blum [Blu94] and Blum and Langley [BL97] proposed the junta-learning problem as a clean formulation of learning in the presence of irrelevant information.

Since the problem was proposed, there has been almost no progress over the naive n^k time algorithm for learning k-juntas under the uniform distribution. The first improvement over the trivial time bound of which we are aware is an unpublished algorithm of A. Kalai and Mansour [KM02] which runs in time roughly $n^{k-\Omega(k^{1/4})}$. Mansour [Man01] later improved this to $n^{k-\Omega(k^{1/2})}$. As stated, our algorithm is the first known superlinear speedup, running in time $n^{.704k}$.

Finally, we note that learning from uniform random examples seems to be the model in which this problem has the right amount of difficulty. Blum and Langley observed [BL97] that if the learning algorithm is allowed to make membership queries then the class of k-juntas can be learned in time $\text{poly}(2^k, n, \log(1/\delta)$. Indeed, this holds true even in the weaker Random Walk model. To see this, we need only show that in the Random Walk model we can check if a particular variable is relevant in time $\text{poly}(n, 2^k, \log(1/\delta))$. (We explicitly show that finding the relevant variables is enough in Proposition 5.5.6.) If the ith variable is relevant to the target function f, then by definition there exists a particular setting y to the k bits of the junta such that $f(y) \neq f(\sigma_i y)$. To check for this, suppose we repeatedly draw $O(n \log n)$ examples to get a to a uniformly random string, and then draw one more example, hoping that the ith bit flips. If it does, *and* if the value of the function also flips, then we conclude that the ith variable is relevant. But also, if the ith variable is indeed relevant then the expected number of times we must repeat the procedure in order to get y to show up and to get the ith bit of y to flip is only $n2^k$. Thus if we repeat the procedure $\mathrm{poly}(n, 2^k, \log(1/\delta))$ times we can test whether the ith variable is relevant with confidence $1 - \delta$.

While membership queries or even the Random Walk model make the problem of learning juntas easy, casting the problem in the more restrictive *statistical query* learning model of Kearns (see [Kea98] for background on this model) makes the problem provably hard. The class of k-juntas over n variables contains at least $\binom{n}{k}$ distinct parity functions, and for any two distinct parity functions $x_S \neq x_T$ we have that $\mathbf{E}[x_S x_T] = 0$. Consequently, an information-theoretic lower bound of Bshouty and Feldman [BF02] implies that *any* statistical query algorithm for learning k-juntas under the uniform distribution must have $q/\tau^2 \geq \binom{n}{k}$, where q is the number of statistical queries the algorithm makes and $\tau \in (0, 1)$ is the additive error tolerance required for each query. Thus improving on the naive running time for learning k-juntas under the uniform distribution in the statistical query model is essentially not possible.

We close this subsection by noting that if we replace the uniform distribution by a product measure in which $\mathbf{P}[x_i = \mathrm{T}] = p_i$, then for almost every choice of $(p_1, \ldots, p_n) \in [0, 1]^n$, the class of k-juntas is learnable in time $\mathrm{poly}(2^k, n, \log(1/\delta))$. In particular, we claim that for every product distribution outside a set of measure zero in $[0, 1]^n$, every k-junta f has nonzero correlation with every variable on which it depends. This easily implies an efficient Fourier-based learning algorithm for identifying all relevant variables, simply by sampling. Our claim is a consequence of the following straightforward fact:

Fact 5.5.1 *If a boolean function f is not a dictator function and f depends on x_i, then $\mathbf{E}_{p_1,\ldots,p_n}[f(x)x_i]$, when viewed formally as a multivariable polynomial in $p_1, \ldots, p_n$, is not identically zero.*

As a consequence, the set of points $(p_1, \ldots, p_n) \in [0, 1]^n$ on which this polynomial takes value 0 has measure 0. The union of all such sets for all (finitely many) choices of i and f

still has measure 0, and the claim is proved.

5.5.2 Representing boolean functions as polynomials

Our learning algorithm for k-juntas will exploit different ways of representing boolean functions as multilinear polynomials. For the remainder of this chapter, instead of always viewing boolean functions as maps $\{+1,-1\}^n \to \{+1,-1\}$, we will take a more abstract view. In general, we will denote an abstract boolean function by $g\colon \{\mathrm{F},\mathrm{T}\}^n \to \{\mathrm{F},\mathrm{T}\}$. We now consider different ways of representing such a g as a multilinear polynomial:

Definition 5.5.2 *Let* $\mathbf{F}$ *be a field and let* $f,t \in \{-1,0,1\}$ *be distinct elements of* $\mathbf{F}$*. We say that a multilinear polynomial* p $\langle \mathbf{F}, f, t\rangle$-represents g *if* $p\colon \mathbf{F}^n \to \mathbf{F}$ *has the following properties:*

- *for all inputs in* $\{f,t\}^n$*,* p *outputs a value in* $\{f,t\}$*; and,*
- p *and* g *induce the same mapping when* F *and* T *are identified with* f *and* t *in the input and output.*

Note that since $f^2, t^2 \in \{0,1\}$, the assumption that p is multilinear is without loss of generality. It is well known that the $\langle \mathbf{F}, f, t\rangle$-representation of g always exists and is unique.

The fields we will consider in this section are the two-element field $\mathbf{F}_2$ and the field $\mathbf{R}$ of real numbers. In $\mathbf{F}_2$ we will represent bits by $f = 0$ and $t = 1$, and in $\mathbf{R}$ we will usually represent bits by $f = +1$, $t = -1$.

Definition 5.5.3 *Given a boolean function* g *on* n *bits,*

- *we write* $g_{\mathbf{F}_2}$ *for the multilinear polynomial which* $\langle \mathbf{F}_2, 0, 1\rangle$*-represents* g*, and we say that* $g_{\mathbf{F}_2}$ $\mathbf{F}_2$-represents g*;*
- *we write* $g_{\mathbf{R}}$ *for the multilinear polynomial which* $\langle \mathbf{R}, +1, -1\rangle$*-represents* g*, and we say that* $g_{\mathbf{R}}$ $\mathbf{R}$-represents g*. Note that this is precisely the Fourier polynomial expansion of* g*.*

As an example, if $g = \mathsf{PARITY}_n$ then we have $g_{\mathbf{F}_2} = x_1 + x_2 + \cdots + x_n$ and $g_{\mathbf{R}} = x_1 x_2 \cdots x_n$. Note that there is a huge difference in the degrees of these two polynomial representations; we will be very interested in the degree of boolean functions under various representations. We observe that for a given field this degree is independent of the exact choice of f, t. This is because we can pass back and forth between any two such choices by nonconstant linear transformations on the inputs and outputs, and under such transformations the monomials of highest degree can never vanish. Thus we can make the following definition:

Definition 5.5.4 $\deg_{\mathbf{F}}(g)$ *is defined to be* $\deg(p)$ *where* p *is any* $\langle \mathbf{F}, f, t\rangle$*-representation of* g.

Hence we have $\deg_{\mathbf{F}_2}(\mathsf{PARITY}_n) = 1$ and $\deg_{\mathbf{R}}(\mathsf{PARITY}_n) = n$. In general $\deg_{\mathbf{F}_2}(g) \leq \deg_{\mathbf{R}}(g)$:

Fact 5.5.5 *For any boolean function* g, $\deg_{\mathbf{F}_2}(g) \leq \deg_{\mathbf{R}}(g)$.

Proof: Let p be the $\langle \mathbf{R}, 0, 1\rangle$-representation of g, so $\deg_{\mathbf{R}}(g) = \deg(p)$. By uniqueness, p must be precisely

$$p(x) = \sum_{z \in \{0,1\}^n} \left[p(z) \left(\prod_{i\colon z_i = 1} x_i \right) \left(\prod_{i\colon z_i = 0} (1 - x_i) \right) \right].$$

But this polynomial plainly has integer coefficients; hence if we reduce the coefficients mod 2 then what we get must be $g_{\mathbf{F}_2}$. This operation can only decrease degree. □

5.5.3 Learning tools for the junta problem

In this subsection we give the learning algorithms we will use for the junta problem. We first show that it suffices to give a learning algorithm which can identify a single relevant variable of an unknown junta. We then give two learning algorithms that look for relevant variables. Our algorithm for learning k-juntas will end up trying both algorithms and we shall prove in Subsection 5.5.4 that at least one of them always works.

Throughout this subsection, f will denote a k-junta on n bits, R will denote the set of variables on which f depends, k' will denote $|R|$ (so $0 \leq k' \leq k$), and f' will denote the function $\{+1, -1\}^{k'} \to \{+1, -1\}$ given by restricting f to R. Recall the definition of *restrictions* of boolean functions, Definition 4.0.7.

Proposition 5.5.6 *Suppose that* $\mathcal{A}$ *is an algorithm running in time* $n^{\alpha} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$ *which can identify at least one variable relevant to* f *with confidence* $1 - \delta$ *(assuming* f *is nonconstant). Then there is an algorithm for exactly learning* f *which runs in time* $n^{\alpha} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$.

Proof: First note that if f is nonconstant then for uniform random inputs each output value occurs with frequency at least 2^{-k}. Hence we can decide whether or not f is a constant function with confidence $1-\delta$ in time $\mathrm{poly}(2^k, n, \log(1/\delta))$.

Next, suppose ρ is any restriction fixing at most k bits. We claim that we can run any learning algorithm on f_ρ with a slowdown of at most $\mathrm{poly}(2^k)$. To do so, we only need to transform the example oracle for f into one for f_ρ; this is easily done by rejecting all samples $\langle x, f(x)\rangle$ for which x does not agree with ρ. Since ρ fixes at most k bits, the probability that a random x agrees with ρ is at least 2^{-k}. Hence with probability $1-\delta$ we can get M samples for f_ρ by taking $M \cdot \mathrm{poly}(2^k)\log(M/\delta)$ samples from the oracle for f.

We now show how to identify all the variables R on which f depends in the requisite amount of time. By induction, suppose we have identified some relevant variables $R' \subseteq R$. For each of the $2^{|R'|}$ possible restrictions ρ which fix the bits in R', consider the function f_ρ. Since f_ρ is also a k-junta, $\mathcal{A}$ can identify some variables relevant to f_ρ (or else we can check that f_ρ is constant). By running $\mathcal{A}$ (with the slowdown described above) for each possible ρ, we will identify new variables to add into R'. We repeatedly add new variables to R', testing all restrictions on these variables, until all of the restricted subfunctions are constant. It is clear that at this point we will have identified all variables relevant to f.

Note that R' grows by at least one variable at each stage, and so we will never run $\mathcal{A}$ more than $k2^k$ times. Further, we can get confidence $1-k^{-1}2^{-k}\delta$ for each run — even after the rejection-sampling slowdown — in time $n^\alpha \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$. Hence we can identify R in time $n^\alpha \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$ with confidence $1-\delta$.

Finally, once R is identified it is easy to learn f exactly. Simply draw $\mathrm{poly}(2^k, \log(1/\delta))$ samples; with probability $1-\delta$ we will see every possible bit setting for R so we can build f's truth table and output this as our hypothesis. □

We now focus on finding algorithms that can identify a random variable given examples from an unknown junta. One technique we have already seen is Fourier coefficient estimation. Note that the Fourier coefficients of f are the same as those of f', and thus each one has a rational value of the form $a/2^k$ for a an integer. Thus using Theorem 5.1.1 we can calculate any Fourier coefficient $\hat{f}(S)$ *exactly* with confidence $1-\delta$ in time $\mathrm{poly}(n, 2^k, \log(1/\delta))$, just by taking $\lambda = 2^{-k-1}$ and rounding the estimate to the nearest multiple of 2^{-k}. Now note that if we ever determine a Fourier coefficient $\hat{f}(S) \neq 0$ with $S \neq \emptyset$, then all variables in S must be relevant to S. This is simply because if f does not depend on $i \in S$ then $\hat{f}(S) = \mathbf{E}[f(x)x_S] = \mathbf{E}[x_i]\mathbf{E}[f(x)x_{S\setminus\{i\}}]$, and $\mathbf{E}[x_i] = 0$. Thus one technique for trying to identifying variables relevant to f is to estimate all of its Fourier coefficients $\hat{f}(S)$ for $1 \leq |S| \leq \alpha$ for some α, looking for one which is nonzero. We immediately get the following:

Proposition 5.5.7 *If $\hat{f'}(S) \neq 0$ for some S with $1 \leq |S| \leq \alpha$, then we can identify at least one relevant variable for f with confidence $1 - \delta$ in time $n^\alpha \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$.*

Our second technique for trying to identify variables relevant to f is to see if f can be represented as a low degree polynomial over the two-element field $\mathbf{F}_2$. A well-known result from computational learning theory [HSW92] says that degree-1 polynomials over $\mathbf{F}_2$ — i.e., parities — can be learned in polynomial time under *any* distribution:

Theorem 5.5.8 *Let $g\colon {\mathbf{F}_2}^N \to \mathbf{F}_2$ be a parity function (i.e., a linear polynomial) on an unknown subset of the N boolean variables $x_1, \ldots, x_N$. There is a learning algorithm $\mathcal{B}$ that, given access to labeled examples $\langle x, g(x)\rangle$ drawn from* any *probability distribution $\mathcal{D}$ on ${\mathbf{F}_2}^N$, outputs a hypothesis h (which is itself a parity of some subset of $x_1, \ldots, x_N$) such that with probability $1 - \delta$ we have $\mathbf{P}_{x\in\mathcal{D}}[h(x) \neq g(x)] \leq \epsilon$. Algorithm $\mathcal{B}$ runs in time $O((\frac{N}{\epsilon} + \frac{\log(1/\delta)}{\epsilon})^\omega)$ where $\omega < 2.376$ is the exponent for matrix multiplication.*

The idea behind Theorem 5.5.8 is simple: since g is a parity function, each labeled example $\langle x, g(x)\rangle$ corresponds to a linear equation over $\mathbf{F}_2$ where the ith unknown corresponds to whether x_i is present in g. Algorithm $\mathcal{B}$ draws $O(\frac{N}{\epsilon} + \frac{\log(1/\delta)}{\epsilon})$ examples and solves the resulting system of linear equations to find some parity over $x_1, \ldots, x_N$ which is consistent with all of the examples. Well-known results in PAC learning theory [BEHW87] imply that such a consistent parity will satisfy the ϵ, δ PAC criterion.

Now suppose $\deg_{\mathbf{F}_2}(f') = \alpha \leq k$. Then f' is a $\mathbf{F}_2$-linear combination (i.e., a parity) over the set of monomials (conjunctions) in $x_1, \ldots, x_n$ of degree up to α. This lets us learn f' in time roughly $n^{\omega\alpha}$:

Proposition 5.5.9 *If $\deg_{\mathbf{F}_2}(f') = \alpha$, then there is a learning algorithm that identifies f exactly in time $n^{\omega\alpha} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$ with confidence $1 - \delta$. (Hence it certainly identifies a variable on which f depends.)*

Proof: Consider the expanded variable space consisting of all monomials over $x_1, \ldots, x_n$ of degree at most α. There are at most $N = n^\alpha$ variables in this space. Run algorithm $\mathcal{B}$

from Theorem 5.5.8 on this variable space, with ϵ set to $2^{-(k+1)}$. That is, given an example $\langle x, f(x)\rangle$, translate it to the example $\langle (x_S)_{|S|\le\alpha}, f(x)\rangle$, and run $\mathcal{B}$ using this new example oracle. Simulating a draw from this new oracle takes time $N\cdot\mathrm{poly}(n)$, so constructing all the necessary examples for $\mathcal{B}$ takes time $N^2\cdot\mathrm{poly}(2^k, n, \log(1/\delta))$. Solving the resulting system of equations takes time $N^\omega\cdot\mathrm{poly}(2^k, n, \log(1/\delta))$. Hence the total time for the algorithm is $n^{\omega\alpha}\cdot\mathrm{poly}(2^k, n, \log(1/\delta))$ as claimed.

We now argue that $\mathcal{B}$'s output hypothesis is precisely the $\mathbf{F}_2$-representation of f. Let $\mathcal{D}$ be the distribution over the expanded variable space induced by the uniform distribution on $x_1, \ldots, x_n$. Since f' (equivalently f) is a parity over the expanded variable space, the output of $\mathcal{B}$ will be a parity hypothesis h over the expanded variable space which satisfies $\mathbf{P}_{x\in\mathcal{D}}[h(x)\ne f(x)] \le 2^{-(k+1)}$.

View both f and h as $\mathbf{F}_2$-polynomials of degree α over the original variables $x_1, \ldots, x_n$. If f and h are not identical, then $f+h \not\equiv 0$ and we have $\mathbf{P}[f(x)\ne h(x)] = \mathbf{P}[f(x)+h(x)\ne 0]$. Now since $\deg_{\mathbf{F}_2}(f+h)\le\alpha$ and $f+h$ is not identically 0, the polynomial $f+h$ must be nonzero on at least a $2^{-\alpha}\ge 2^{-k}$ fraction of the points in ${\mathbf{F}_2}^n$, by the Schwartz-Zippel Lemma. But this contradicts the fact that $\mathbf{P}_{x\in\mathcal{D}}[h(x)\ne f(x)]\le 2^{-(k+1)}$. □

5.5.4 Learning using new structural properties of boolean functions

With our learning tools in hand we are ready to give the algorithm for learning k-juntas. The basic idea is to show that every boolean function f' must either have a nonzero Fourier coefficient of "not too large" positive degree, or must be a polynomial over $\mathbf{F}_2$ of "not too large" degree. Then by Propositions 5.5.7 and 5.5.9, in either case we can find a relevant variable for f' without performing a full-fledged exhaustive search.

The Fourier learning algorithm described earlier fails only on functions whose low-degree Fourier coefficients are all zero (except for possibly the constant coefficient; if this is nonzero the Fourier algorithm can still fail). Let us make a definition for such functions:

Definition 5.5.10 *Suppose that g satisfies $\hat{g}(S) = 0$ for all $1\le|S|<t$. If $\hat{g}(\emptyset)$ is also* 0 *then we say that g is* strongly balanced up to size t. *If $\hat{g}(\emptyset)$ is nonzero we say that g is* strongly biased up to size t.

These definitions were essentially first made by Bernasconi in [Ber01]. The justification of the terminology is this: if g is strongly balanced up to size t, then it is easy to show that every subfunction of g obtained by fixing $0 \leq \ell \leq t-1$ bits is balanced. Similarly, if g is strongly biased up to size t then it is easy to show that every such subfunction has the same bias as g itself.

We now show that strongly balanced functions have low $\mathbf{F}_2$-degree:

Theorem 5.5.11 *Let $g \notin \{\mathsf{PARITY}_n, -\mathsf{PARITY}_n\}$ be a boolean function on n bits which is strongly balanced up to size t. Then $\deg_{\mathbf{F}_2}(g) \leq n-t$.*

Proof: Given such a g, let $h = g \oplus \mathsf{PARITY}_n$. Then $h_{\mathbf{R}} = g_{\mathbf{R}} \cdot x_1 x_2 \cdots x_n$. By assumption, $g_{\mathbf{R}}$ has zero coefficient on all monomials x_S with $|S| < t$. By multilinear reduction ($x_i^2 = 1$) we see that $h_{\mathbf{R}}$ has zero coefficient on all monomials x_S with $|S| > n-t$. Hence $\deg_{\mathbf{R}}(h) \leq n-t$, so by Fact 5.5.5, $\deg_{\mathbf{F}_2}(h) \leq n-t$. But since $g = h \oplus \mathsf{PARITY}_n$, the $\mathbf{F}_2$-representation of g is simply $g_{\mathbf{F}_2}(x) = h_{\mathbf{F}_2}(x) + x_1 + \cdots + x_n$. Adding a degree-1 polynomial to $h_{\mathbf{F}_2}$ does not increase degree (since g is neither PARITY_n nor its negation, h is not a constant function and hence $\deg_{\mathbf{F}_2}(h) \geq 1$), and consequently $\deg_{\mathbf{F}_2}(g) \leq n-t$. □

The bound $n-t$ in Theorem 5.5.11 is best possible. To see this, consider the function

$$g(x) = (x_1 \wedge \cdots \wedge x_{n-t}) \oplus x_{n-t+1} \oplus \cdots \oplus x_n.$$

This function has $\mathbf{F}_2$-representation $g_{\mathbf{F}_2}(x) = x_1 \cdots x_{n-t} + x_{n-t+1} + \cdots + x_n$ so $\deg_{\mathbf{F}_2}(g) = n-t$. Moreover, g is balanced and every subfunction of g fixing fewer than t bits is also balanced, since to make g unbalanced one must restrict all of $x_{n-k+1}, \ldots, x_n$.

It remains to deal with strongly biased functions. Our next theorem shows that no boolean function can be strongly biased up to too large a size:

Theorem 5.5.12 *If g is a nonconstant boolean function on n bits which is strongly biased up to size t, then $t \leq \frac{2}{3}n$.*

Proof: Let $g_{\mathbf{R}}(x) = \sum_S c_S x_S$ be the $\mathbf{R}$-representation of g. Since g is nonconstant and strongly biased up to size t we have $0 < |c_\emptyset| < 1$ and $c_S = 0$ for all $0 < |S| < t$. As in Theorem 5.5.11, we let $h = g \oplus \mathsf{PARITY}_n$ so $h_{\mathbf{R}}(x) = c_\emptyset x_1 x_2 \cdots x_n + \sum_{|S| \leq n-t} c'_S x_S$, where $c'_S = c_{[n] \setminus S}$.

Let $h' \colon \{+1,-1\}^n \to \{1+c_\emptyset, 1-c_\emptyset, -1+c_\emptyset, -1-c_\emptyset\}$ be the real-valued function given by $h'(x) = h_{\mathbf{R}}(x) - c_\emptyset x_1 x_2 \cdots x_n$; note that $\deg(h') \leq n-t$. Furthermore, for $x \in \{+1,-1\}^n$ we have $h'(x) \in \{1+c_\emptyset, 1-c_\emptyset\}$ if and only if $h_{\mathbf{R}}(x) = +1$, and $h'(x) \in \{-1+c_\emptyset, -1-c_\emptyset\}$ if and only if $h_{\mathbf{R}}(x) = -1$. Since $0 < |c_\emptyset| < 1$ we have that $\{1+c_\emptyset, 1-c_\emptyset\}$ and $\{-1+c_\emptyset, -1-c_\emptyset\}$ are disjoint two-element sets.

Let $p\colon \mathbf{R} \to \mathbf{R}$ be the degree 3 polynomial which maps $1+c_\emptyset$ and $1-c_\emptyset$ to $+1$ and $-1-c_\emptyset$ and $-1+c_\emptyset$ to -1. Now consider the polynomial $p \circ h'$. By construction $p \circ h'$ maps $\{+1,-1\}^n \to \{+1,-1\}$, and $p \circ h'$ $\mathbf{R}$-represents h. But the $\mathbf{R}$-representation of h is unique, so after multilinear reduction $p \circ h'$ must be identical to $h_{\mathbf{R}}$. Since $c_\emptyset \neq 0$, we know that $\deg_{\mathbf{R}}(h)$ is exactly n. Since p has degree exactly 3 and $\deg(h') \leq n-t$, we conclude that $3(n-t) \geq n$, whence $t \leq \frac{2}{3}n$. □

Following publication of the previous result, Regev [Reg03] gave a quicker proof of Theorem 5.5.12 which we now present:

Proof: (Regev) Let $g_{\mathbf{R}}(x)$ be as in the previous proof. Let U be any set of maximal size such that $c_U \neq 0$; since g is nonconstant and strongly biased up to size t we have $|U| \geq t$. Consider expanding $g_{\mathbf{R}}(x)^2 = (\sum_S c_S x_S)(\sum_T c_T x_T)$; there will be a nonzero coefficient on the cross-term $x_\emptyset x_U$. But $g_{\mathbf{R}}(x)^2$ must be identically 1 after multilinear reduction, since $g_{\mathbf{R}}$ takes on only the values ±1. Thus the nonzero coefficient on x_U must be cancelled in the expansion. But if $t > \frac{2}{3}n$ then $c_T = 0$ for all $1 \leq |T| \leq \frac{2}{3}n$, and by the pigeonhole principle all nonzero cross-terms not involving the constant term $c_\emptyset$ will be on terms x_V with $V < \frac{2}{3}n$. This contradicts the fact that the x_U term must be cancelled. □

The bound $\frac{2}{3}n$ in Theorem 5.5.12 is best possible. To see this, let $n = 3m$ and consider the function

$$f(x_1, \ldots, x_n) = \left(\bigoplus_{i=1}^{2m} x_i\right) \bigwedge \left(\bigoplus_{i=m+1}^{n} x_i\right).$$

It is easy to see that this function is unbalanced, and also that its bias cannot change under any restriction of fewer than $2m$ bits (to change the bias, one must set bits $1 \ldots 2m$ or $m+1 \ldots 3m$ or $1 \ldots m, 2m+1 \ldots 3m$).

We can now prove our main theorem:

Theorem 5.5.13 *The class of k-juntas over n bits can be exactly learned under the uniform distribution with confidence $1-\delta$ in time $n^{\frac{\omega}{\omega+1}k} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$.*

Proof: Let f be a k-junta on n bits and f' be the function on at most k bits given by restricting f to its relevant variables. Let $t = \frac{\omega}{\omega+1}k > \frac{2}{3}k$. If f' is strongly balanced up to size t then by Theorem 5.5.11 f' is an $\mathbf{F}_2$-polynomial of degree at most $k - t = k/(\omega + 1)$. By Proposition 5.5.9 f' can be learned in time $(n^{k/(\omega+1)})^{\omega} \cdot \text{poly}(2^k, n, \log(1/\delta))$. On the other hand, suppose f' is not strongly balanced up to size t. By Theorem 5.5.12, f' cannot be strongly biased up to size t, since $t > \frac{2}{3}k$. Hence f' has a nonzero Fourier coefficient of degree less than t and greater than 0. So by Proposition 5.5.7, some relevant variable for f can be identified in time $n^t \cdot \text{poly}(2^k, n, \log(1/\delta))$.

In either case, we can identify some relevant variable for f in the claimed time, $n^{\frac{\omega}{\omega+1}k} \cdot \text{poly}(2^k, n, \log(1/\delta))$. Proposition 5.5.6 completes the proof. □

5.5.5 Variants of the junta learning problem

We can use the ideas developed thus far to analyze some variants and special cases of the juntas learning problem. We begin by describing various subclasses of k-juntas for which the learning problem is more easily solved:

Monotone juntas: Every variable x_i relevant to f' has strictly positive influence, in the sense of Definition 2.3.4: i.e., $I_i(f) > 0$. But when f' is monotone, $I_i(f) = \hat{f}'(\{i\})$. Thus the class of monotone k-juntas can be learned in time $\text{poly}(2^k, n, \log(1/\delta))$ using the Fourier learning algorithm of Proposition 5.5.7.

Random juntas: As observed in [BL97], almost every k-junta on n variables can be learned in time $\text{poly}(2^k, n, \log(1/\delta))$. To see this, observe that if a function f' on k bits is chosen uniformly at random, then for every S we have $\hat{f}'(S) = 0$ only if exactly half of all inputs have $f'(x) = x_S$. This occurs with probability $\binom{2^k}{2^{k-1}}/2^{2^k} = O(1)/2^{k/2}$. Consequently, with overwhelming probability in terms of k — at least $1 - O(k)/2^{k/2}$ — a random function on k variables will have every Fourier coefficient of degree 1 nonzero, and hence we can learn using Proposition 5.5.7.

Symmetric juntas: A *symmetric* k-junta is a junta whose value depends only on how many of its k relevant variables are set to true. We can learn the class of symmetric k-juntas in time $n^{\frac{2}{3}k} \cdot \text{poly}(2^k, n, \log(1/\delta))$, which is a slight improvement on our bound for arbitrary k-juntas. To prove this, we show that every symmetric function f' on k variables other than parity and its negation has a nonzero Fourier coefficient $\hat{f}'(S)$ for $1 \leq |S| < \frac{2}{3}k$.

Hence we can identify at least one relevant variable in time $n^{\frac{2}{3}k} \cdot \mathrm{poly}(2^k, n, \log(1/\delta))$ using Proposition 5.5.7, and we can use the algorithm of Proposition 5.5.6 since the class of symmetric functions is closed under subfunctions.

To prove this claim about the Fourier coefficients of symmetric functions, first note that if f' is not balanced then by Theorem 5.5.12 it must have a nonzero Fourier coefficient of positive degree less than $\frac{2}{3}k$. Otherwise, if f' is balanced and is neither parity nor its negation, then the function $g = f' \oplus \mathsf{PARITY}_k$ is a symmetric nonconstant function and $\deg_{\mathbf{R}}(g) < k$; this last fact follows because the $x_1 x_2 \cdots x_k$ coefficient of g is the constant coefficient of f', and f' is balanced. By a result of von zur Gathen and Roche [vzGR97], every nonconstant symmetric function g on k variables has $\deg_{\mathbf{R}}(g) \geq k - O(k^{.548})$. Hence $\hat{g}(S) \neq 0$ for some $k - O(k^{.548}) \leq |S| < k$, so $\hat{f'}([k] \setminus S) \neq 0$ and $1 \leq |[k] \setminus S| \leq O(k^{.548}) \leq \frac{2}{3}k$.

By considering the work of von zur Gathen and Roche it seems probable that every symmetric boolean function on k bits has a nonzero Fourier coefficient on a much smaller nonzero degree; perhaps $O(\log k)$ or even $O(1)$. But a proof would be at least as difficult as improving von zur Gathen and Roche's conjecture that every symmetric nonconstant function has degree at least $k - O(1)$.

Chapter 6

Coin Flipping From A Cosmic Source

As we saw in Section 2.4, the study of the noise sensitivity of boolean functions is equivalent to the study of the 2-norm of the Bonami-Beckner operator T_ρ. In this chapter we introduce a new problem in theoretical computer science which essentially amounts to studying the higher norms of this operator.

6.1 Coin flipping from a cosmic source

Consider a "cosmic" source of truly random bits $x \in \{+1,-1\}^n$ which is accessible to k distributed parties. If the k parties want to use the source to obtain a common single random bit, they can easily do so by deciding beforehand to let the common bit be x_1. More generally, they can decide beforehand on any balanced function $f\colon \{+1,-1\}^n \to \{+1,-1\}$ and let the common bit be $f(x)$.

In this setting, there is no real advantage in taking the function f to be anything other than the dictator function $f = \pi_n^1$. The problem becomes more interesting when the parties independently receive *noisy* versions of the cosmic random bits. That is, party i receives the string y^i and the strings y^i are independently distributed as $N_\epsilon(x)$. Assume the parties are separated and cannot communicate; however they still want to toss the same fair coin given their noisy versions of the source. We will now allow each party i to use a different balanced function $f_i\colon \{+1,-1\}^n \to \{+1,-1\}$ as a coin tossing procedure. We want to maximize the probability that all of the parties agree on their coin flips; i.e., $\mathbf{P}[f_1(y^1) = \cdots = f_k(y^k)]$.

This problem has natural motivations in cryptography. One setting in which it very relevant is that of the "everlasting security" protocol of Ding and Rabin [DR02]. This protocol gives a very secure encryption algorithm in the bounded storage model. However, the authors presuppose the existence of a satellite or other cosmic source which broadcasts a continuous stream of a huge number of random bits — on the order of a trillion per second. It is natural to expect that the distributed parties using this source will have some reception errors. Further, it is not clear that putting an error-correcting code on the bits is feasible or even desirable — the parties may not trust the performer of the encoding, and may prefer instead to agree on a measurable random source from nature. Another cryptographic problem somewhat related to our cosmic coin flipping problem was studied by Maurer [Mau97].

This cosmic coin flipping problem is also of interest as a noncryptographic collective coin flipping problem. Another example of such a problem is the full information model, introduced by Ben-Or and Linial [BL90] and studied extensively (see, e.g., the brief survey in [Dod00] and the references therein). In this problem, many parties try to agree on a single random bit; each generates a random coin toss, and there is a single protocol (function) taking all the coin tosses and producing a bit. The difficulty arises from the assumption that some parties are corrupt and can choose their coins adversarially. In our problem, the major difference is that the parties do not communicate any random bits, so they each must apply a protocol to a shared random string. And, instead of arbitrary corruptions, we assume random ones.

The problem we study in this chapter is also a natural question regarding error correction for the broadcast channel (see, e.g., [CT68]) with a truly random source. Naturally, when the source is truly random, error correction is impossible. However, here instead of requiring that every party obtains all of the information transmitted with high probability, we only require that all parties attain some mutual information with high probability, and that this mutual information has high entropy.

Finally, this problem is a strict generalization of the study of the noise sensitivity of balanced functions. To see this, suppose $k = 2$ and both parties use the same balanced function f. Then the first party's bits y^1 are uniformly random, and the second party's bits y^2 can be viewed as being distributed as $N_{\epsilon'}(y^1)$, where $\epsilon' = 2\epsilon(1-\epsilon)$. Thus the success probability of the parties is precisely $1 - \mathsf{NS}_{\epsilon'}(f)$.

As mentioned previously, for $k > 2$ the problem can be viewed as the study of the higher norms of the Bonami-Beckner operator. Suppose again that all parties use the same function f for their coin-tossing protocols, and that f has the property that, when using it, the parties are equally likely to agree on $+1$ and -1. (See Theorem 6.5.1 and the discussion of antisymmetry in the next section for more on these assumptions.) Then by symmetry we may as well compute twice the probability that the parties agree on -1. By Fact 2.4.2, for every source string x and for each i independently, the probability that party i outputs -1 is precisely $\frac{1}{2} - \frac{1}{2}T_{1-2\epsilon}(f)(x) = T_{1-2\epsilon}(\frac{1}{2} - \frac{1}{2}f)(x)$. Write $f' \colon \{+1,-1\}^n \to \{0,1\}$ for the boolean function $\frac{1}{2} - \frac{1}{2}f$. Then the overall probability of success is $2\mathbf{E}_x[(T_{1-2\epsilon}(f')(x))^k] = 2||T_{1-2\epsilon}(f')||_k^k$. This justifies our statement that the problem studied in this chapter is equivalent to the problem of studying the higher norms of the Bonami-Beckner operator T_ρ.

6.2 Definitions and notation

Let us define the problem precisely:

The model: *Let $k \geq 1$ be the number of* parties*, let $n \geq 1$ be the* block length *and let $\epsilon \in (0, \frac{1}{2})$ be the* corruption probability*. Our probability space is the space all sequences $(x, y^1, \ldots, y^k) \in \{+1,-1\}^{n(k+1)}$. Here x represents the* source *and is chosen uniformly at random from $\{+1,-1\}^n$; y^i represents the bits that party i holds and is distributed as $N_\epsilon(x)$, independently for each i. When we write $\mathbf{P}$ and $\mathbf{E}$ we mean probability and expected value in this space.*

Balanced and antisymmetric functions: *Let $\mathcal{B}_n$ denote the set of* balanced *boolean functions $f \colon \{+1,-1\}^n \to \{+1,-1\}$; i.e., those satisfying $\mathbf{E}[f] = 0$. Let $\mathcal{A}_n$ denote the set of* antisymmetric *boolean functions $f \colon \{+1,-1\}^n \to \{+1,-1\}$; i.e., those satisfying $f(-x) = -x$. Note that $\mathcal{A}_n \subset \mathcal{B}_n$.*

Protocols: *A* protocol *consists of k functions $f_i \in \mathcal{B}_n$. An* antisymmetric protocol *consists of k functions $f_i \in \mathcal{A}_n$. For a protocol $(f_1, \ldots, f_k)$ we write $\mathcal{P}(f_1, \ldots, f_k; \epsilon)$ for the probability that all parties agree when using $f_1, \ldots, f_k$, so*

$$\mathcal{P}(f_1, \ldots, f_k; \epsilon) = \mathbf{P}[f_1(y^1) = \cdots = f_k(y^k)].$$

We write $\mathcal{P}_k(f;\epsilon)$ in place of $\mathcal{P}(f_1,\ldots,f_k;\epsilon)$ in case $f_1 = \cdots = f_k = f$.

It turns out that restricting all f_i to be balanced is neither necessary nor sufficient for ensuring that the output bit, when agreed upon, is uniformly random. Non-necessity is not particularly interesting; e.g., we may have $k = n = 2$, $f_1 = \mathsf{AND}_2$, $f_2 = \mathsf{OR}_2$. More interesting is non-sufficiency; see Proposition 6.10.4 for a counterexample, requiring $k \geq 3$, $n \geq 5$. A sufficient condition is that every function be antisymmetric, since in this case,

$$\begin{aligned}\mathbf{P}[f_1(y^1) = \cdots = f_k(y^k) = +1] &= \mathbf{P}[f_1(-y^1) = f_k(-y^k) = -1] \\ &= \mathbf{P}[f(y^1) = \cdots = f(y^k) = -1],\end{aligned}$$

where the first equality is by antisymmetry and the second since $\mathbf{P}$ assigns the same probability to $(x, y^1, \ldots, y^k)$ as it does to $(-x, -y^1, \ldots, -y^k)$. We are not aware of a weaker condition than antisymmetry that ensures that the output bit, when agreed upon, is uniformly random. In any case, we shall consider both antisymmetric and merely balanced protocols.

We end this section with a few more definitions. For $S \subseteq [n]$ and π a permutation of $[n]$, let $\pi_S \colon \{+1,-1\}^n \to \{+1,-1\}^n$ be defined by $\pi_S(x)_i = x_{\pi(i)}$ if $i \in S$ and $\pi_S(x)_i = -x_{\pi(i)}$ if $i \notin S$. Any π_S merely permutes coordinates and switches the roles of ± 1 on some coordinates. It is therefore easy to see that $\mathcal{P}(f_1 \circ \pi_S, \ldots, f_k \circ \pi_S; \epsilon) = \mathcal{P}(f_1, \ldots, f_k; \epsilon)$ for any π_S.

In order to express uniqueness results cleanly, we abuse language in the following way: For particular k, n, and ϵ, we say that $(f_1, \ldots, f_k)$ is the unique best protocol "up to π_S" if the set of best protocols is exactly $\{(f_1 \circ \pi_S, \ldots, f_k \circ \pi_S) \colon S \subseteq [n], \pi \in S_n\}$. This notation should not be confused with the notation for a dictator function π_n^i.

6.3 Outline of results

In this section we outline our main results, which we prove in the succeeding sections.

One reason we study the probability that all parties agree on a random bit — rather than an alternative metric such as the expected number of pairs of parties who agree or the expected number of parties in the majority — is that these alternative metrics are subsumed by the case $k = 2$. In this case, and also in case $k = 3$, Fourier analysis gives an exact solution to our problem, and the best protocol up to π_S is for all parties to use the first-bit dictator function, $f = \pi_n^1$. We attribute the case $k = 2$ in the following theorem to folklore:

Theorem 6.4.1 *For all k, n, ϵ, if we wish to maximize the expression*

$$\mathbf{E}[\#(i,j)\colon f_i(y^i) = f_j(y^j)], \tag{6.1}$$

the unique best protocol up to π_S is given by $f_1 = \cdots = f_k = \pi_n^1$. In particular, if $k = 2$ or $k = 3$, then for all n and ϵ the unique best protocol, up to π_S, for maximizing $\mathcal{P}(f_1, \ldots, f_k; \epsilon)$ is given by $f_1 = \cdots = f_k = \pi_n^1$.

In general we do not know how to find the optimal protocol for every n, k, and ϵ. However, we can prove some general properties of the protocols which maximize $\mathcal{P}(f_1, \ldots, f_k; \epsilon)$. First we show that, as might be expected, in any optimal protocol all parties use the same function:

Theorem 6.5.1 *Fix k, n, and ϵ. Let $\mathcal{C}$ be any class of boolean functions on n bits. Subject to the restriction that $f_1, \ldots, f_k \in \mathcal{C}$, every protocol which maximizes $\mathcal{P}(f_1, \ldots, f_k; \epsilon)$ has $f_1 = \cdots = f_k$.*

The proof of this uses convexity.

Knowing that all parties should use the same function, we can derive further information about what this function must look like. Recall from Section 3.6 that Benjamini et al. showed that shifting a function to make it monotone causes its noise sensitivity to go down. The same phenomenon occurs in the present problem. Via such shiftings we can show that the best protocol is a monotone function. Indeed, we can say more. For $x, y \in \{+1, -1\}^n$, let us write $x \preceq_L y$ if $\sum_{i=1}^m x_i \leq \sum_{i=1}^m y_i$ for every $m = 1 \ldots n$. We call a function f *left-monotone* if $f(x) \leq f(y)$ whenever $x \preceq_L y$. (This partial ordering has been studied in other contexts; for example, left-monotone functions are equivalent to shifted simplicial complexes [Kli03].) Note that the partial order induced by $\preceq_L$ is a refinement of the usual partial order on the hypercube, and in particular, every left-monotone function is monotone. By shifting again, we have the following:

Theorem 6.6.1 *For all k, n, and ϵ, if f maximizes $\mathcal{P}_k(f; \epsilon)$ among all protocols then*

f is left-monotone (up to π_S). This theorem remains true if "protocol" is replaced by "antisymmetric protocol".

So far we have not ruled out the possibility that the optimal protocol always consists of taking just one bit. However when n and ϵ are fixed and $k \to \infty$, it becomes better to use the majority of *all* the bits (rather than just the first). Using a coupling argument, we prove the following result:

Theorem 6.7.2 *For all n odd and all ϵ, there exists a $K = K(n, \epsilon)$ such that for $k \geq K$, the unique best protocol up to π_S is given by $f = \mathsf{MAJ}_n$. Moreover, as $k \to \infty$,*

$$\mathcal{P}_k(\mathsf{MAJ}_n; \epsilon) = \Theta\left((1 - \mathbf{P}[\mathrm{Bin}(n, \epsilon) > n/2])^k\right),$$

where $\mathrm{Bin}(n, \epsilon)$ is a binomial variable with parameters n and ϵ. (This should be compared to $\Theta((1-\epsilon)^k)$ for the protocol $f = \pi_n^1$.) When n is even, a similar result is true; in place of MAJ_n, one should take any balanced function f which has $f(x) = +1$ whenever $\sum_{i=1}^n x_i > 0$ and $f(x) = -1$ whenever $\sum_{i=1}^n x_i < 0$.

A dual result is obtained by fixing n and k, and letting ϵ go to either 0 or $\frac{1}{2}$. The proof uses isoperimetry and Fourier analysis.

Theorem 6.8.1 *For all k and n, there exist $0 < \epsilon' = \epsilon'(n,k) < \epsilon'' = \epsilon''(n,k) < \frac{1}{2}$ such that for all $0 < \epsilon < \epsilon'$ or $\epsilon'' < \epsilon < \frac{1}{2}$, the unique best protocol up to π_S is given by the dictator function $f = \pi_n^1 = \mathsf{MAJ}_1$.*

It may now seem like the optimal protocol consists of either taking all functions to be MAJ_1 or all functions to be MAJ_n. This is not the case however, as a computer-assisted proof shows that sometimes MAJ_r is better than MAJ_1 and MAJ_n for $1 < r < n$. See Proposition 6.10.2.

Finally we come to what is perhaps the most interesting asymptotic setting for n, k, and ϵ; namely, ϵ a fixed constant in $(0, \frac{1}{2})$, $k \to \infty$, and n an unbounded function of k and ϵ. In this case we are unable to determine the best protocol function. However, we are able to show an asymptotically tight result. Despite Theorem 6.7.2, it is not true that as $k \to \infty$, the success probability of the best protocol goes to 0 exponentially fast in k. If the parties use the majority protocol on a large number of bits, their success probability goes to 0 only as a slowly diminishing polynomial in k; furthermore, this protocol is essentially the best possible.

Theorem 6.9.1 *Fix $\epsilon \in (0, \frac{1}{2})$ and let $\nu = \nu(\epsilon) = \frac{4\epsilon(1-\epsilon)}{(1-2\epsilon)^2}$. Then as $k \to \infty$,*

$$\sup_{f \in \mathcal{B}} \mathcal{P}_k(f; \epsilon) = \tilde{\Theta}\left(k^{-\nu}\right),$$

where the supremum is taken over all balanced boolean functions f on finite numbers of bits, and $\tilde{\Theta}(\cdot)$ denotes asymptotics to within a subpolynomial ($k^{o(1)}$) factor. The upper bound is achieved asymptotically by the majority function MAJ_n with n sufficiently large (in terms of k and ϵ).

The proof of the upper bound uses the reverse Bonami-Beckner inequality.

Finally, it is natural to ask if the optimal function is always MAJ_r for some $1 \le r \le n$ (assuming, say, n is odd). We conjecture that this is the case, at least for antisymmetric protocols:

Conjecture 6.3.1 *For any k, ϵ, and odd n, there is an odd $1 \le r \le n$ such that $\mathcal{P}_k(f; \epsilon)$ is maximized among antisymmetric functions by $f = \mathsf{MAJ}_r$.*

In fact, we know of no counterexample to Conjecture 6.3.1 even if we allow the parties to use any balanced function (which could allow for a biased output). Some evidence that resolving this conjecture could possibly be hard: One, it is not true that for any non-majority function f, and any fixed k, there is a majority function which dominates f over all ϵ — see Proposition 6.10.3 for a computer-assisted counterexample. Two, for certain k and ϵ, $\mathcal{P}_k(\mathsf{MAJ}_r, \epsilon)$ is not even unimodal as a function of r. For example, one can show by explicit calculation (as described in Section 6.10) that for $k = 12$ and $\epsilon = .1$, we have $\mathcal{P}_k(\mathsf{MAJ}_1; \epsilon) \ge \mathcal{P}_k(\mathsf{MAJ}_3; \epsilon)$, $\mathcal{P}_k(\mathsf{MAJ}_3; \epsilon) \le \mathcal{P}_k(\mathsf{MAJ}_5; \epsilon) \le \cdots \le \mathcal{P}_k(\mathsf{MAJ}_{11}; \epsilon)$, and $\mathcal{P}(\mathsf{MAJ}_{11}; \epsilon) \ge \mathcal{P}(\mathsf{MAJ}_{13}; \epsilon)$, and it appears as though the sequence is decreasing from this point on.

In the following sections we prove the theorems given above, and close with a section describing some computer-assisted analysis of the problem.

6.4 $k = 2, 3$

In this section we show that for $k = 2, 3$, all parties should use the same function. The reason, roughly speaking, is that when $k = 2$ and the parties use the same function, the problem reduces to finding the *least* noise-sensitive balanced functions; as we know from Proposition 3.2.1, these functions are precisely the dictator functions. Only a few more ideas are needed to deal with the possibility the parties use different function, and with the case $k = 3$.

Recall Theorem 6.4.1:

Theorem 6.4.1 *For all k, n, ϵ, if we wish to maximize the expression*

$$\mathbf{E}[\#(i,j)\colon f_i(y^i) = f_j(y^j)], \tag{6.2}$$

the unique best protocol up to π_S is given by $f_1 = \cdots = f_k = \pi_n^1$. In particular, if $k = 2$ or $k = 3$, then for all n and ϵ the unique best protocol, up to π_S, for maximizing $\mathcal{P}(f_1, \dots, f_k; \epsilon)$ is given by $f_1 = \cdots = f_k = \pi_n^1$.

Proof: Let us analyze the quantity (6.2):

$$\begin{aligned}
\mathbf{E}[\#(i,j)\colon f_i(y^i) = f_j(y^j)] &= \sum_{1 \le i < j \le k} \mathbf{E}[\mathbf{1}_{f_i(y^i) = f_j(y^j)}] \\
&= \sum_{i<j} \mathbf{E}\left[\frac{1}{2} + \frac{1}{2} f_i(y^i) f_j(y^j)\right] \\
&= \frac{1}{2}\binom{k}{2} + \frac{1}{2}\sum_{i<j} \mathbf{E}[f_i(y^i) f_j(y^j)].
\end{aligned}$$

Using the same argument as in the proof of Proposition 2.3.1, it is easy to show that

$$\mathbf{E}[f_i(y^i) f_j(y^j)] = \sum_{S \subseteq [n]} (1 - 2\epsilon)^{2|S|} \hat{f}_i(S) \hat{f}_j(S).$$

Note that this sum may be taken over $S \neq \emptyset$, since f_i and f_j are assumed balanced. Now by Cauchy-Schwarz,

$$\begin{aligned}\sum_{S\neq\emptyset}(1-2\epsilon)^{2|S|}\hat{f}_i(S)\hat{f}_j(S) &\leq \left[\sum_{S\neq\emptyset}(1-2\epsilon)^{2|S|}\hat{f}_i(S)^2\right]^{\frac{1}{2}}\left[\sum_{S\neq\emptyset}(1-2\epsilon)^{2|S|}\hat{f}_j(S)^2\right]^{\frac{1}{2}} \\ &= \left(1-2\mathsf{NS}_{2\epsilon(1-\epsilon)}(f_i)\right)^{\frac{1}{2}}\left(1-2\mathsf{NS}_{2\epsilon(1-\epsilon)}(f_j)\right)^{\frac{1}{2}} \\ &\leq (1-2\epsilon)^2,\end{aligned}$$

where the second inequality is by Proposition 3.2.1. This inequality is tight if and only if f_i and f_j are both dictator functions. In this case, the first inequality is tight if and only if f_i and f_j are the *same* dictator function. Since any dictator function is the function π_n^1 up to π_S, the proof of the first part of the theorem is complete.

For the second part, first note that in the case $k = 2$, the quantity $\mathcal{P}(f_1, f_2; \epsilon)$ is precisely $\mathbf{E}[\#(i,j)\colon f_i(y^i) = f_j(y^j)]$, since only $i = 1$, $j = 2$ is possible. In the case $k = 3$, whenever all parties agree there are three pairs who agree, and otherwise there is exactly one pair who agrees. Hence $\mathcal{P}(f_1, f_2, f_3; \epsilon) = \frac{1}{2}\mathbf{E}[\#(i,j)\colon f_i(y^i) = f_j(y^j)] - \frac{1}{2}$, so maximizing $\mathcal{P}(f_1, f_2, f_3; \epsilon)$ is equivalent to maximizing (6.2). □

6.5 All parties should use the same function

In this section we show that, as might be expected, in any optimal protocol all parties use the same function:

Theorem 6.5.1 *Fix k, n, and ϵ. Let $\mathcal{C}$ be any class of boolean functions on n bits. Subject to the restriction that $f_1, \dots, f_k \in \mathcal{C}$, every protocol which maximizes $\mathcal{P}(f_1, \dots, f_k; \epsilon)$ has $f_1 = \cdots = f_k$.*

Proof: Let $\mathcal{C} = \{f_1, f_2, \dots, f_M\}$, and assume $M > 1$, else the theorem is trivial. Assume $\mathcal{C}$ doesn't contain one of the constant functions, else again the theorem is trivial. Now suppose that among the k parties, exactly t_i use the function f_i. Then clearly,

$$t_i \geq 0, \qquad \sum_{i=1}^{M} t_i = k, \qquad t_i \in \mathbf{Z}. \tag{6.3}$$

For each i, write $c_i(x)$ for the probability that a party outputs $+1$ given that the source string is x. By Fact 2.4.2, $c_i(x) = \frac{1}{2} + \frac{1}{2}T_{1-2\epsilon}(f)(x)$. Writing $\mathcal{P} = \mathcal{P}(f_1, \dots, f_k; \epsilon)$ for short, we have exactly

$$\mathcal{P} = \sum_{x \in \{+1,-1\}^n} 2^{-n} \left(\prod_{i=1}^{M} (c_i(x))^{t_i} + \prod_{i=1}^{M} (1 - c_i(x))^{t_i} \right).$$

Since no f_i is constant we have $c_i(x) \in (0,1)$ for all i. Note that for any $c \in (0,1)$, the function $g(t) = c^t$, is log-convex (since $\log c^t = t \log c$ is linear). Therefore the function $g_1 \cdots g_M \colon \mathbf{R}^M \to \mathbf{R}$ given by $(t_1, \ldots, t_M) \mapsto \prod_{i=1}^{M} g_i(t_i)$ is a log-convex function, and therefore a convex function. Since the sum of convex functions is also convex, we conclude that $\mathcal{P}$ is a convex function of the t_i's. We wish to maximize $\mathcal{P}$ subject to the restrictions (6.3).

If we relax the assumption $t_i \in \mathbf{Z}$ to $t_i \in \mathbf{R}$, we are simply maximizing a convex function over a convex bounded polytope. The vertices of the polytope are simply the points of the form $(0, \ldots, 0, k, 0, \ldots, 0)$. The maximum must occur at a vertex, and so it follows that there is at least one maximizing protocol in which all parties use the same function.

It remains to show that $\mathcal{P}$ does not obtain a maximum at any point which is not a vertex of the polytope. Note that by convexity, if $\mathcal{P}$ has a maximum which is not a vertex of the polytope, then there exists an interval $I = \{\lambda v_1 + (1-\lambda) v_2 \colon \lambda \in [0,1]\}$, where v_1 and v_2 are vertices of the polytope, such that f is a constant function on I. Therefore if we could show that f is strictly convex on I (as a function of λ), then it will follow that the maximum is obtained only at vertices of the polytope.

Note that when restricted to the edge I joining, for example, $v_1 = (k, 0, \ldots, 0)$ and $v_2 = (0, k, 0, \ldots, 0)$, $\mathcal{P}$ is given by

$$\sum_{x \in \{+1,-1\}^n} 2^{-n} \left((c_1(x))^{\lambda k} (c_2(x))^{(1-\lambda)k} + (1 - c_1)(x))^{\lambda k} (1 - c_2(x))^{(1-\lambda)k} \right).$$

Since $T_{1-2\epsilon}$ is a 1-1 operator on the space of functions $\{+1,-1\}^n \to \mathbf{R}$ by Fact 2.4.2, there must be some $x_0 \in \{+1,-1\}^n$ such that $T_{1-2\epsilon}(f_1)(x_0) \neq T_{1-2\epsilon}(f_2)(x_0)$ and hence $c_1(x_0) \neq c_2(x_0)$. Using $c_1(x_0) \neq c_2(x_0)$ it is easy to show that

$$(c_1(x_0))^{\lambda k} (c_2(x_0))^{(1-\lambda)k} = (c_2(x_0))^k \left(\frac{c_1(x_0)}{c_2(x_0)} \right)^{\lambda k}$$

is a strictly convex function of λ. Thus $\mathcal{P}$ is strictly convex on I, as needed. □

6.6 The protocol should be left-monotone

Because of the results in the previous section, we will assume from now on that the parties all use the same function. In this section we shall show that the success probability is maximized when the parties' function is left-monotone (up to π_S):

Theorem 6.6.1 *For all k, n, and ϵ, if f maximizes $\mathcal{P}_k(f;\epsilon)$ among all protocols, then f is left-monotone (up to π_S). This theorem remains true if the phrase "protocol" is replaced by "antisymmetric protocol".*

We will prove this in two parts; first we shall show in Proposition 6.6.2 that the shifting operators κ_i increase the probability of success; this suffices to show that f must be monotone up to π_S (i.e., f must be unate). We then define a left-shifting operator and show in Proposition 6.6.4 that it increases the success probability of monotone functions. Since repeated left-shifting will produce a left-monotone function, this will show that f must be left-monotone up to π_S. The two propositions taken together prove Theorem 6.6.1.

Proposition 6.6.2 *Let $\mathcal{C}$ stand for either $\mathcal{B}_n$ or A_n. For any k, n, ϵ, if f is restricted to be in $\mathcal{C}$, then $\mathcal{P}_k(f;\epsilon)$ is maximized only if f is monotone up to π_S; i.e., f is unate.*

Proof: Let $f \in \mathcal{C}$ be any function which maximizes $\mathcal{P}_k(f;\epsilon)$ among functions in $\mathcal{C}$. Recall the shifting operators κ_i from Definition 3.6.1. We shall show that $\mathcal{P}_k(\kappa_i f;\epsilon) \geq \mathcal{P}_k(f;\epsilon)$, with equality only if f is either monotone or anti-monotone in the ith coordinate. Since shifting on all coordinates produces a monotone function, this will complete the proof.

Without loss of generality assume $i = 1$, and write $f' = \kappa_1 f$, so for each $x \in \{+1,-1\}^{n-1}$,

- if $f(-1,x) = f(+1,x)$ then $f'(-1,x) = f'(+1,x) = f(-1,x) = f(+1,x)$;
- if $f(-1,x) \neq f(+1,x)$ then $f'(-1,x) = -1$, $f'(+1,x) = +1$.

It is easy to see that in the case $\mathcal{C} = \mathcal{B}_n$, f' remains in $\mathcal{C}$; a little thought reveals that this is again true in the case $\mathcal{C} = \mathcal{A}_n$.

For $y \in \{+1,-1\}^n$, let $\tilde{y} \in \{+1,-1\}^{n-1}$ denote the last $n-1$ bits of y. To show $\mathcal{P}_k(f';\epsilon) \geq \mathcal{P}(f;\epsilon)$, we will show that for all $z^1,\ldots,z^k \in \{+1,-1\}^{n-1}$,

$$\begin{aligned}\mathbf{P}[f'(y^1) = \cdots = f'(y^k) \mid \tilde{y}^1 = z^1,\ldots,\tilde{y}^k = z^k]&\\ \geq \mathbf{P}[f(y^1) = \cdots = f(y^k) \mid \tilde{y}^1 = z^1,\ldots,\tilde{y}^k = z^k].& \quad (6.4)\end{aligned}$$

So suppose each y^i's last $n-1$ bits are fixed to be z^i. Given z^i, $f(y^i)$ is a function from $\{+1,-1\}$ to $\{+1,-1\}$, and is therefore either a constant function ± 1 or a dictator function $\pm\pi^1$.

If $f(y^i)$ is already determined by z^i, then so is $f'(y^i)$ and the determined value is the same. Otherwise, $f(y^i)$ is a nonconstant function of the one remaining unknown bit, y^i_1, and is thus either π^1 or $-\pi^1$. In *either* case, $f'(y^i)$ is the identity function π^1 on y^i_1

Assume that given $(z^1,\ldots,z^k)$, there are $a+b$ undetermined functions $f(y^i_1)$, with a of them π^1, and b of them $-\pi^1$. The probability that all of these functions agree on $+1$ (or -1) is

$$q = \frac{1}{2}\left((1-\epsilon)^a\epsilon^b + \epsilon^a(1-\epsilon)^b\right),$$

and the probability that all of the undetermined f''s agree on $+1$ (or -1) is

$$q' = \frac{1}{2}\left((1-\epsilon)^{a+b} + \epsilon^{a+b}\right).$$

There are three cases to consider:

- If some of the determined functions are determined to be $+1$ and some to be -1, then both terms in (6.4) are zero.
- If all of the determined functions are determined to be $+1$ (-1), then the left side of (6.4) is q' and the right side of (6.4) is q.
- If there are no determined functions, then the left side of (6.4) is $2q'$ and the right side of (6.4) is $2q$.

Therefore our claim that $\mathcal{P}_k(f';\epsilon) \geq \mathcal{P}_k(f;\epsilon)$ will follow once we show that $q' \geq q$. But this is

$$\begin{aligned} \frac{1}{2}(1-\epsilon)^{a+b} + \epsilon^{a+b} &\geq \frac{1}{2}(1-\epsilon)^a\epsilon^b + \epsilon^a(1-\epsilon)^b \\ \Leftrightarrow 1 + \left(\frac{\epsilon}{1-\epsilon}\right)^{a+b} &\geq \left(\frac{\epsilon}{1-\epsilon}\right)^b + \left(\frac{\epsilon}{1-\epsilon}\right)^a, \end{aligned} \tag{6.5}$$

which follows by the convexity of the function $t \to \left(\frac{\epsilon}{1-\epsilon}\right)^t$.

Thus we've established $\mathcal{P}_k(f';\epsilon) \geq \mathcal{P}(f;\epsilon)$. It remains to prove that this inequality is strict unless f was already monotone or anti-monotone on the first coordinate. If f is neither monotone nor anti-monotone on the first coordinate, then there exist z^1 and z^2 such that f, when the last $n-1$ coordinates are restricted to z^1, becomes π^1, and when the last $n-1$ coordinates are restricted to z^2, becomes $-\pi^1$. Picking $z^3, \ldots, z^k$ so that all the other restricted functions are $\pm\pi^1$, we obtain $a, b \geq 1$, so (6.5) is strict inequality and therefore $q' > q$. □

To show that it is best for the protocol to be left-monotone, we introduce a left-shift operator for monotone functions:

Definition 6.6.3 *Given a monotone $f \colon \{+1,-1\}^n \to \{+1,-1\}$ and $i \neq j \in [n]$, we define the operator λ_{ij} which maps f to another monotone function $\lambda_{ij} f$ as follows: For each x, consider the function on two bits gotten by restricting f to have input x except on coordinates i and j. Since f is monotone, there are only 6 possibilities for what the restricted function is; its support may be $\emptyset$, $\{(-1,-1)\}$, $\{(-1,-1),(-1,+1)\}$, $\{(-1,-1),(+1,-1)\}$, $\{(-1,-1),(-1,+1),(+1,-1)\}$ or $\{(-1,-1),(-1,+1),(+1,-1),(+1,+1)\}$. Define $\lambda_{ij} f$ to be the same function in all cases except when the support is $\{(-1,-1),(+1,-1)\}$; in this case, switch it to $\{(-1,-1),(-1,+1)\}$.*

It is straightforward to check that λ_{ij} preserves balance and antisymmetry and that applying all possible left-shifts to a monotone function produces a left-monotone function.

Proposition 6.6.4 *Let $\mathcal{C}$ stand for either $\mathcal{B}_n$ or A_n. For any k, n, and ϵ, if f is restricted to be in $\mathcal{C}$, then any function which maximizes $\mathcal{P}_k(f;\epsilon)$ must be left-monotone, up to π_S.*

Proof: The proof is similar to the proof of Proposition 6.6.2, so we proceed briefly. By Proposition 6.6.2, we may assume that any maximizing f is monotone. We again show that for all $i < j$, $\mathcal{P}_k(\lambda_{ij} f;\epsilon) \geq \mathcal{P}_k(f;\epsilon)$, with equality only if f is already left-shifted on coordinates i, j.

Write $f' = \lambda_{ij} f$. As before, we condition on all but the i and j bits of each $y_1, \ldots, y_k$, and show that f' is better in every case. Say that under this conditioning, a of the $f(y^i)$'s restrict to the function with support $\{(-1,-1),(-1,+1)\}$, and b of the $f(y^i)$'s restrict to the function with support $\{(-1,-1),(+1,-1)\}$. Since all other possible restricted functions have the same value for $(+1,-1)$ as they do for $(-1,+1)$, it suffices again to compare the probability with which the $a+b$ functions agree on -1 with the probability that the corresponding shifted functions agree on -1. Further, by symmetry, we need only consider the cases when the two source bits from x are different (otherwise f and f' do equally well).

So considering the two cases — the source bits are $(-1,+1)$ or the source bits are $(+1,-1)$ — we get that the contribution from the f-restricted functions will be $\frac{1}{2}((1-\epsilon)^a\epsilon^b) + \epsilon^b(1-\epsilon)^a)$, and the contribution from their shifted versions will be $\frac{1}{2}((1-\epsilon)^{a+b} + \epsilon^{a+b})$. As we saw in Proposition 6.6.2, this latter quantity is always at least the former quantity. Hence the shift can only improve the probability of agreement.

Thus we indeed have $\mathcal{P}_k(f';\epsilon) \geq \mathcal{P}_k(f;\epsilon)$. Now note that if none of the shifting operations strictly increased the probability of agreement for f, then for every pair of coordinates (i,j) which were shifted, either all the balanced restrictions of f to coordinates (i,j) had support $\{(-1,-1),(-1,+1)\}$, or all the balanced restrictions had support $\{(-1,-1),(+1,-1)\}$. In either case, all the shifting did was replace the function f by a function $f \circ \pi_\emptyset$, where $\pi_\emptyset$ is the transposition of coordinates (i,j). It thus follows that the original function was already left-monotone up to some $\pi_\emptyset$, as needed. □

6.7 Fixed ϵ, n; $k \to \infty$

In this section we consider the case that ϵ and n are fixed and $k \to \infty$. In this case, for sufficiently large k the best protocol is the majority of all the bits, MAJ_n. The intuitive reason is as follows: When the number of parties is extremely large — equivalently, when considering $||T_\rho||_k$ for extremely large k — one would rather have a protocol f which is extremely stable at a few inputs, rather than one which is moderately stable on all inputs. Thus the majority function MAJ_n, which is extremely stable to noise when the source input is $+\vec{1} = (+1,\ldots,+1)$ or $-\vec{1} = (-1,\ldots,-1)$, is preferable to the dictator function π^1, which is $1-\epsilon$ stable at all inputs. We proceed to make this intuition rigorous.

Given a boolean function f, let us write $p_+(f,x,\epsilon)$ for the probability that $f(N_\epsilon(x)) = +1$ and $p_-(f,x,\epsilon)$ for the probability that $f(N_\epsilon(x)) = -1$.

Lemma 6.7.1 *Fix ϵ and let f be monotone. Then as a function of x, $p_+(f,x,\epsilon)$ is maximized at $x = +\vec{1}$, and $p_-(f,x,\epsilon)$ is maximized at $x = -\vec{1}$.*

Proof: We prove the lemma for p_1, the proof for p_0 being the same. Recalling Definition 2.1.1, we think of the noise operator as acting by *updating* each bit with probability

2ϵ. Let $x \in \{+1,-1\}^n$ be any string. Let $x' = N_\epsilon(x)$ and $+\vec{1}' = N_\epsilon(+\vec{1})$. We claim that we can couple the random variables x' and $+\vec{1}'$ in such a way that $x' \leq +\vec{1}'$ (in the sense of the monotone partial order). The coupling is achieved in the easiest possible way: we update the same bits of x and $+\vec{1}$ with the same values. Clearly, we have $x' \leq +\vec{1}'$. Hence by monotonicity, if $f(x') = +1$ then $f(+\vec{1}') = +1$. Thus

$$p_+(f,x,\epsilon) = \mathbf{P}[f(x') = +1] \leq \mathbf{P}[f(+\vec{1}') = +1] = p_+(f,+\vec{1},\epsilon),$$

as needed. □

Theorem 6.7.2 *For all constant odd n and all ϵ, there exists a $K = K(n,\epsilon)$ such that for $k \geq K$, the unique best protocol up to π_S is given by $f = \mathsf{MAJ}_n$. Moreover, as $k \to \infty$,*

$$\mathcal{P}_k(\mathsf{MAJ}_n;\epsilon) = \Theta\left((1 - \mathbf{P}[\mathrm{Bin}(n,\epsilon) > n/2])^k\right),$$

where $\mathrm{Bin}(n,\epsilon)$ *is a binomial variable with parameters n and ϵ. (This should be compared to $\Theta((1-\epsilon)^k)$ for the protocol $f = \pi_n^1$.) When n is even, a similar result is true; in place of* MAJ_n*, one should take any balanced function f which has $f(x) = +1$ whenever $\sum_{i=1}^n x_i > 0$ and $f(x) = -1$ whenever $\sum_{i=1}^n x_i < 0$.*

Proof: For simplicity we will prove the theorem only for odd n; the case of even n is simply wordier. By Theorems 6.5.1 and 6.6.1, we may assume without loss of generality that all parties use the same monotone function $f \in \mathcal{B}_n$. Now on one hand,

$$\begin{aligned}\mathcal{P}_k(\mathsf{MAJ}_n;\epsilon) &= 2^{-n} \sum_{x\in\{+1,-1\}^n} (p_+(\mathsf{MAJ}_n,x,\epsilon)^k + p_-(\mathsf{MAJ}_n,x,\epsilon)^k) \\ &\geq 2^{-n}(p_+(\mathsf{MAJ}_n,+\vec{1},\epsilon)^k + p_-(\mathsf{MAJ}_n,-\vec{1},\epsilon)^k). \qquad (6.6)\end{aligned}$$

On the other hand, using Lemma 6.7.1,

$$\begin{aligned}\mathcal{P}_k(f;\epsilon) &= 2^{-n} \sum_{x\in\{+1,-1\}^n} (p_+(f,x,\epsilon)^k + (p_-(f,x,\epsilon))^k) \\ &\leq 2^{-n} \sum_{x\in\{+1,-1\}^n} (p_+(f,+\vec{1},\epsilon)^k + p_-(f,-\vec{1},\epsilon)^k) \\ &= p_+(f,+\vec{1},\epsilon)^k + p_-(f,-\vec{1},\epsilon)^k. \qquad (6.7)\end{aligned}$$

We claim that if $f \neq \mathsf{MAJ}_n$ then both $p_+(f, +\vec{1}, \epsilon) < p_+(\mathsf{MAJ}_n, +\vec{1}, \epsilon)$ and $p_-(f, -\vec{1}, \epsilon) < p_-(\mathsf{MAJ}_n, -\vec{1}, \epsilon)$. By comparing (6.6) and (6.7) one can see that if this is true, we will indeed have $\mathcal{P}_k(f; \epsilon) < \mathcal{P}_k(\mathsf{MAJ}_n; \epsilon)$ for sufficiently large k.

To show the claim, simply note that

$$p_+(f, +\vec{1}, \epsilon) = \sum_{x \in f^{-1}(+1)} (1-\epsilon)^{n-\Delta(x,+\vec{1})} \epsilon^{\Delta(x,+\vec{1})}, \tag{6.8}$$

where Δ denotes Hamming distance. The quantity being summed is a strictly decreasing function of $\Delta(x, +\vec{1})$. Thus if $f \in \mathcal{B}_n$ differs from MAJ_n then we will indeed have $p_+(f, +\vec{1}, \epsilon) < p_+(\mathsf{MAJ}_n, +\vec{1}, \epsilon)$. The proof for p_- is similar.

The proof is now complete except for the asymptotic bound given in the statement of the theorem. This bound follows immediately from (6.6), (6.7), and (6.8) once we note that $p_+(\mathsf{MAJ}_n, +\vec{1}, \epsilon) = p_-(\mathsf{MAJ}_n, -\vec{1}, \epsilon) = 1 - \mathbf{P}[\mathrm{Bin}(n, \epsilon) > n/2]$. □

6.8 Fixed k, n; $\epsilon \to 0$ or $\epsilon \to \frac{1}{2}$

We now consider different asymptotics for the parameters; k and n are fixed arbitrarily and $\epsilon \to 0$ or $\epsilon \to \frac{1}{2}$. In both cases, the best protocol is the dictator function, but separate proofs are required. When $\epsilon \to 0$, the intuition is that the probability of failure is dominated by the case when there is only a single error among all the parties; hence we want the function which is most stable to a single error. Isoperimetry shows this is the dictator function. When $\epsilon \to \frac{1}{2}$, the intuition is just the opposite; we imagine that among all the bits the parties hold, there are two parties with correlated bits on some coordinate, and otherwise everything is uniformly random. In this case, Fourier analysis shows that the dictator function is again the best.

Theorem 6.8.1 *For all k and n, there exist $0 < \epsilon' = \epsilon'(n,k) < \epsilon'' = \epsilon''(n,k) < \frac{1}{2}$ such that for all $0 < \epsilon < \epsilon'$ or $\epsilon'' < \epsilon < \frac{1}{2}$, the unique best protocol up to π_S is given by the dictator function $f = \pi_n^1$.*

We prove the two halves of the theorem separately.

Proposition 6.8.2 *For all k and n, there exists $\epsilon'(k,n) > 0$ such that for all $0 < \epsilon < \epsilon'(k,n)$, the unique best protocol up to π_S is given by the dictator function $f = \pi_n^1$.*

Proof: As $\epsilon \to 0$, the probability that there is more than one corrupted bit among the players is $O(\epsilon^2)$ (here the constant in the $O(\cdot)$ *does* depend on k and n). Suppose that there is exactly one flipped bit, and it is bit i for some party j. Then all the parties will agree if and only if $f(x) = f(\sigma_i x)$. We therefore obtain the following:

$$\mathcal{P}_k(f;\epsilon) = (1-\epsilon)^{kn} + k\epsilon(1-\epsilon)^{kn-1}\sum_{i=1}^{n}\mathbf{P}_x[f(x) = f(\sigma_i x)] + O(\epsilon^2).$$

Thus for ϵ sufficiently small (compared to k and n), maximizing $\mathcal{P}_k(f;\epsilon)$ is equivalent to maximizing $\sum_{i=1}^{n}\mathbf{P}_x[f(x) = f(\sigma_i x)]$.

Write A for the subset of the hypercube $\{x\colon f(x) = -1\}$, and $\partial_E(A)$ for the edge-boundary of the set A,

$$\partial_E(A) = \cup_{i=1}^{n}\{(x,\sigma_i x)\colon x \in A, \sigma_i x \notin A\}.$$

It's easily seen that

$$\sum_{i=1}^{n}\mathbf{P}_x[f(x) = f(\sigma_i x))] = n - 2^{-n+1}|\partial_E(A)|.$$

Thus the maximizing balanced protocol f is the one which minimizes $|\partial_E(A)|$ over sets A such that $|A| = 2^{n-1}$. By the edge-isoperimetric inequality for the cube, the minimizing sets A are precisely those of the form $\{x\colon x_i = -1\}$, or $\{x\colon x_i = +1\}$. Thus the best protocol f is π_n^1 up to π_S, as claimed. $\Box$

Proposition 6.8.3 *For all k and n, there exists $\epsilon'(k,n) < \frac{1}{2}$ such that for all $\epsilon'(k,n) < \epsilon < \frac{1}{2}$, the unique best protocol up to π_S is given by the dictator function $f = \pi_n^1$.*

Proof: The proof is somewhat similar to the previous one. Let us think of the noise operator N_ϵ as *updating* bits with probability 2ϵ. We will imagine the tableau of bits the parties receive as being generated in the following manner. For $1 \le i \le k$ and $1 \le j \le n$, define X_{ij} to be i.i.d. random variables taking 1 with probability $\delta = 1 - 2\epsilon$ and 0 with probability δ. We then let x be a uniformly random string, and we define y^i as follows: $y^i_j = x_j$ if $X_{ij} = 1$, and otherwise y^i_j is chosen uniformly at random.

When $\epsilon \to \frac{1}{2}$, $\delta \to 0$, so almost all of the X_{ij}'s will be 0. If *all* of the $X(i,j)$'s are 0 then the parties' strings are uniform and independent. Thus since f is balanced,

$$\mathbf{P}[f(y^1) = \cdots = f(y^k) \mid \sum_{i,j} X_{ij} = 0] = 2^{-k+1}.$$

Indeed, this can be true even if not all X_{ij}'s are 0. Suppose that $\sum X_{ij} = 1$. Then the parties' bits are still uniform and independent. (The single correlation with x is irrelevant, since x is uniform.) Hence

$$\mathbf{P}[f(y^1) = \cdots = f(y^k) \mid \sum_{i,j} X_{ij} = 1] = 2^{-k+1}.$$

Even more, if $\sum X_{ij} = 2$ but it is because $X_{ij} = X_{i'j'} = 1$ where $j \neq j'$, then the parties' bits are uniform and independent for the same reason:

$$\mathbf{P}[f(y^1) = \cdots = f(y^k) \mid X_{ij} = X_{i'j'} = 1, \sum_{s,t} X_{s,t} = 2] = 2^{-k+1}, \text{ for } j \neq j'.$$

For $\sum X_{ij} = 2$, there is only effective correlation among the parties' strings if we have $X_{ij} = X_{i'j} = 1$ for some $i \neq i'$. Now note that

$$\mathbf{P}\left[\sum_{i,j} X_{ij} > 2\right] = O(\delta^3)$$

(where the constant in the $O(\cdot)$ *does* depend on k and n).

We therefore conclude

$$\mathcal{P}_k(f;\epsilon) = C_{k,n,\delta} + \delta^2(1-\delta)^{kn-2} \sum_{i<i'} \sum_{j=1}^{n} \mathbf{P}[f(y^1) = \cdots = f(y^k) \mid X_{i,j} = X_{i',j} = 1] + O(\delta^3), \tag{6.9}$$

where $C_{k,n,\delta}$ is independent of f. Thus for $\epsilon \to \frac{1}{2}$ — i.e., $\delta \to 0$ — we see that maximizing $\mathcal{P}_k(f;\epsilon)$ over balanced functions is equivalent to maximizing

$$\sum_{i<i'} \sum_{j=1}^{n} \mathbf{P}[f(y^1) = \cdots = f(y^k) \mid X_{i,j} = X_{i',j} = 1]. \tag{6.10}$$

Write z for a uniformly random string from $\{+1,-1\}^n$, and z' for a string which is chosen by first picking $1 \leq j \leq n$ uniformly at random, and then choosing $z' \in \{+1,-1\}^n$ uniformly among all z' such that $z'_j = z_j$. Then it is easy to see that (6.10) is equal to

$$n\binom{k}{2}\mathbf{P}[f(z) = f(z')]2^{-k+1},$$

since f is balanced and the values of i and i' are irrelevant. Thus the maximizing protocols are those f maximizing $\mathbf{P}[f(z) = f(z')] = \frac{1}{2} + \frac{1}{2}\mathbf{E}[f(z)f(z')]$. We pass to the Fourier representation of f. It is very easy to check that

$$\mathbf{E}[z_S z'_T] = \begin{cases} 0 & \text{if } S \neq T, \\ 0 & \text{if } S = T,\ |S| > 1, \\ \frac{1}{n} & \text{if } S = T,\ |S| = 1, \\ 1 & \text{if } S = T = \emptyset. \end{cases}$$

It follows that $\mathbf{E}[f(z)f(z')] = \hat{f}(\emptyset)^2 + \frac{1}{n}\sum_{|S|=1}\hat{f}(S)^2$. But $\hat{f}(\emptyset)^2 = 0$ since f is balanced, and $\frac{1}{n}\sum_{|S|=1}\hat{f}(S)^2 \leq \frac{1}{n}$, with equality if and only if f has Fourier degree 1. By Proposition 3.2.1 this happens if and only if f is a dictator function. The proof is complete. □

6.9 Fixed ϵ; $k \to \infty$ with n unbounded

Finally, we come to the most natural asymptotic question, and the most difficult: if we treat ϵ as fixed and allow n to be unbounded (in terms of k and ϵ), what is the best protocol as $k \to \infty$, and what success rate does it achieve?

Unlike in the previous two sections, we do not know the best protocol in this case. However we *are* able to give an asymptotically good answer. In this section we show that the optimal success probability decreases as a slowly diminishing polynomial, and that the majority protocol on many bits is asymptotically near-best:

Theorem 6.9.1 *Fix $\epsilon \in (0, \frac{1}{2})$ and let $\nu = \nu(\epsilon) = \frac{4\epsilon(1-\epsilon)}{(1-2\epsilon)^2}$. Then as $k \to \infty$,*

$$\sup_{f \in \mathcal{B}} \mathcal{P}_k(f;\epsilon) = \tilde{\Theta}\left(k^{-\nu}\right),$$

where the supremum is taken over all balanced boolean functions f on finite numbers of bits, *and $\tilde{\Theta}(\cdot)$ denotes asymptotics to within a subpolynomial ($k^{o(1)}$) factor. The upper bound is achieved asymptotically by the majority function MAJ_n with n sufficiently large (in terms of k and ϵ).*

We shall prove Theorem 6.9.1 in two parts; first we estimate the success probability of the majority protocol $\lim_{n\to\infty} \mathcal{P}_k(\mathsf{MAJ}_n; \epsilon)$ and show it is $\Omega(k^{-\nu})$. We then show this is asymptotically best possible.

Theorem 6.9.2 *Fix $\epsilon \in (0, \frac{1}{2})$. Then*

$$\lim_{n\to\infty} \mathcal{P}_k(\mathsf{MAJ}_n; \epsilon) \geq \Omega(k^{-\nu}).$$

In particular, there is a sequence (n_k) with $n_k = O(k^{2\nu})$ for which the above bound holds.

Proof: We begin by establishing

$$\lim_{\substack{n\to\infty\\ n \text{ odd}}} \mathcal{P}_k(\mathsf{MAJ}_n; \epsilon) = \frac{2\nu^{\frac{1}{2}}}{(2\pi)^{\frac{1}{2}(\nu-1)}} \int_0^1 t^k I(t)^{\nu-1}\, dt, \tag{6.11}$$

where $I = \phi \circ \Phi^{-1}$ is the so-called Gaussian isoperimetric function, with ϕ and Φ the density and distribution functions of a standard normal random variable.

Apply Lemma 3.4.1, with $X \sim N(0,1)$ representing $n^{-\frac{1}{2}}$ times the sum of the bits in the cosmic source, and $Y|X \sim N((1-2\epsilon)X, 4\epsilon(1-\epsilon))$ representing $n^{-\frac{1}{2}}$ times the sum of the bits in any given party's string. Thus as $n \to \infty$, the probability that all parties agree on $+1$ when using MAJ_n is precisely

$$\int_{-\infty}^{\infty} \Phi\left(\frac{(1-2\epsilon)x}{[4\epsilon(1-\epsilon)]^{\frac{1}{2}}}\right)^k \phi(x)\, dx.$$

By symmetry we can multiply this by 2 to get the probability that all k parties agree. Making the change of variables $t = \Phi(\nu^{-\frac{1}{2}}x)$, $x = \nu^{\frac{1}{2}}\Phi^{-1}(t)$, $dx = \nu^{\frac{1}{2}} I(t)^{-1}\, dt$, we get

$$\begin{aligned}\lim_{n\to\infty} \mathcal{P}_k(\mathsf{MAJ}_n; \epsilon) &= 2\nu^{\frac{1}{2}} \int_0^1 z^k \phi(\nu^{\frac{1}{2}}\Phi^{-1}(t)) I(t)^{-1}\, dt \\ &= \frac{2\nu^{\frac{1}{2}}}{(2\pi)^{\frac{1}{2}(\nu-1)}} \int_0^1 t^k I(t)^{\nu-1}\, dt,\end{aligned}$$

as claimed.

We now estimate the integral in (6.11). It is known (see, e.g., [BG99]) that $I(t) \geq J(t(1-t))$, where $J(t) = t\sqrt{\ln(1/t)}$. We will forego the marginal improvements given by taking the logarithmic term and simply use the estimate $I(t) \geq t(1-t)$. We then get

$$\begin{aligned}
\int_0^1 t^k I(t)^{\nu-1}\,dt &\geq \int_0^1 t^k (t(1-t))^{\nu-1} \\
&= \frac{\Gamma(\nu)\Gamma(k+\nu)}{\Gamma(k+2\nu)} && \text{([AS72, 6.2.1, 6.2.2])} \\
&\geq \Gamma(\nu)(k+2\nu)^{-\nu} && \text{(Stirling approximation).}
\end{aligned}$$

Substituting this estimate into (6.11) we get $\lim_{n\to\infty} \mathcal{P}_k(\mathsf{MAJ}_n;\epsilon) \geq c(\nu)k^{-\nu}$ for some $c(\nu) > 0$ depending only on ϵ, as desired. By the error bound from Proposition 3.4.1, the lower bound holds with a smaller constant as long as n is at least as large as $O(k^{2\nu})$. $\square$

The formula (6.11) can be used to get very accurate estimates of the majority protocol's probability of success. For example, if we take $\epsilon = \frac{1}{2} - \frac{\sqrt{2}}{4}$ so that $\nu = 1$ then we get $\lim_{n\to\infty} \mathcal{P}_k(\mathsf{MAJ}_n;\epsilon) = \frac{2}{k+1}$. A combinatorial explanation of this fact would be interesting.

We now complete the proof of Theorem 6.9.1 by showing an upper bound of $k^{-\nu+o(1)}$ for the success probability of k parties.

Lemma 6.9.3 *Let $f, g\colon \{+1,-1\}^n \to \mathbf{R}^{\geq 0}$ be nonnegative, let $x \in \{+1,-1\}^n$ be chosen uniformly at random, and let $y = N_\epsilon(x)$, where $\epsilon \in [0, \frac{1}{2}]$. Then for $p, q \leq 1$,*

$$\mathbf{E}[f(x)g(y)] \geq ||f||_p||g||_q \qquad \textit{whenever } (1-p)(1-q) \geq (1-2\epsilon)^2.$$

Proof: Write $\rho = 1 - 2\epsilon$. By Definition 2.4.1, $\mathbf{E}[f(x)g(y)] = \mathbf{E}[f \cdot T_\rho(g)]$. Since f and $T_\rho(g)$ are nonnegative and $\frac{1}{p} + \frac{p-1}{p} = 1$, we can apply the reverse Hölder inequality to conclude $\mathbf{E}[f \cdot T_\rho(g)] \geq ||f||_p||T_\rho(g)||_{\frac{p}{p-1}}$. Now apply the reverse Bonami-Beckner inequality, Theorem 2.4.9, to conclude $||T_\rho(g)||_{\frac{p}{p-1}} \geq ||g||_{1-\frac{\rho^2}{1-p}}$. But $1 - p \geq \frac{\rho^2}{1-q}$, so $1 - \frac{\rho^2}{1-p} \geq q$ and hence $||g||_{1-\frac{\rho^2}{1-p}} \geq ||g||_q$, as desired. $\square$

Theorem 6.9.4 *Let $f\colon \{+1,-1\}^n \to \{0,1\}$ and suppose $\mathbf{E}[f] \leq \frac{1}{2}$. Then for any fixed $\epsilon \in (0, \frac{1}{2})$, as $k \to \infty$, $||T_{1-2\epsilon}f||_k^k \leq k^{-\nu+o(1)}$.*

Proof: Write $\rho = 1 - 2\epsilon$. Suppose we have $\mathbf{E}[(T_\rho f)^k] \geq 2\delta$ for some δ. Define $G = \{x \in \{+1,-1\}^n \mid (T_\rho f)(x)^k \geq \delta\}$. Then since $0 \leq (T_\rho f)^k \leq 1$, we conclude that $|G| \geq \delta 2^{-n}$. Let g be the 0-1 indicator function for G. Now on one hand,

$$\begin{aligned}
\langle T_\rho(1-f), g\rangle &= \langle 1, g\rangle - \langle T_\rho f, g\rangle \\
&\leq \delta - \langle T_\rho f, g\rangle \\
&\leq \delta - \delta \cdot \delta^{\frac{1}{k}},
\end{aligned}$$

the last inequality following because $(T_\rho f)(x) \geq \delta^{\frac{1}{k}}$ for all x with $g(x) = 1$. On the other hand, if we apply Lemma 6.9.3 with $p = \log^{-\frac{1}{2}} k$ and $q = 1 - \frac{1}{1-p}\rho^2 = 1 - (1+o(1))\rho^2$, we get the following:

$$\begin{aligned}
\langle T_\rho(1-f), g\rangle &\geq ||1-f||_p ||g||_{1-(1+o(1))\rho^2} \\
&\geq \left(\frac{1}{2}\right)^{\frac{1}{p}} \delta^{\frac{1}{1-(1+o(1))\rho^2}} \\
&= k^{-o(1)} \delta^{(1+o(1))\frac{1}{1-\rho^2}}.
\end{aligned}$$

Thus

$$\begin{array}{rrcl}
 & \delta - \delta \cdot \delta^{\frac{1}{k}} & \geq & k^{-o(1)}\delta^{(1+o(1))\frac{1}{1-\rho^2}} \\
\Rightarrow & 1 - k^{-o(1)}\delta^{(1+o(1))(\frac{\rho^2}{1-\rho^2})} & \geq & \delta^{\frac{1}{k}} \\
\Rightarrow & \ln\left(1 - k^{-o(1)}\delta^{(1+o(1))(\frac{\rho^2}{1-\rho^2})}\right) & \geq & \frac{1}{k}\ln(\delta) \\
\Rightarrow & -k^{-o(1)}\delta^{(1+o(1))(\frac{\rho^2}{1-\rho^2})} & \geq & \frac{1}{k}\ln(\delta) \\
\Rightarrow & \dfrac{\delta^{(1+o(1))(\frac{\rho^2}{1-\rho^2})}}{\ln(1/\delta)} & \leq & \dfrac{1}{k^{1-o(1)}} \\
\Rightarrow & \delta & \leq & k^{-\frac{1-\rho^2}{\rho^2}+o(1)} \\
\Rightarrow & 2\delta & \leq & k^{-\nu+o(1)},
\end{array}$$

which completes the proof. □

Corollary 6.9.5 *Fix $\epsilon \in (0, \frac{1}{2})$. Then for all balanced functions $f \colon \{+1,-1\}^n \to \{+1,-1\}$, the success probability $\mathcal{P}_k(f;\epsilon)$ is at most $k^{-\nu+o(1)}$.*

Proof: As argued at the end of Section 6.1, the probability that all k parties agree on -1 when using f is precisely $||T_{1-2\epsilon}(\frac{1}{2} - \frac{1}{2}f)||_k^k$. By symmetry, the probability that all k parties agree on $+1$ is $||T_{1-2\epsilon}(\frac{1}{2} + \frac{1}{2}f)||_k^k$. Since $\frac{1}{2} \pm \frac{1}{2}f$ are 0-1 functions with expectation exactly

f	$\mathcal{P}_k(f;\epsilon)$
MAJ_1	$\epsilon^k + (1-\epsilon)^k$
T_1	$1/16\,(-6\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5+4\,\epsilon^2)^k+1/16\,(1+6\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5-4\,\epsilon^2)^k+1/16\,(4\,\epsilon-10\,\epsilon^2+10\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+1/16\,(1-4\,\epsilon+10\,\epsilon^2-10\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+1/4\,(\epsilon-\epsilon^2+4\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+1/4\,(1-\epsilon+\epsilon^2-4\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+1/4\,(1-2\,\epsilon+4\,\epsilon^2-6\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+1/4\,(2\,\epsilon-4\,\epsilon^2+6\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+3/8\,(\epsilon+\epsilon^2-4\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+3/8\,(1-\epsilon-\epsilon^2+4\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k$
T_2	$1/8\,(-2\,\epsilon^3+3\,\epsilon^2)^k+1/8\,(1+2\,\epsilon^3-3\,\epsilon^2)^k+1/8\,(3\,\epsilon-6\,\epsilon^2+4\,\epsilon^3)^k+1/8\,(1-3\,\epsilon+6\,\epsilon^2-4\,\epsilon^3)^k+3/8\,\epsilon^k+3/8\,(1-\epsilon)^k+3/8\,(1-2\,\epsilon+3\,\epsilon^2-2\,\epsilon^3)^k+3/8\,(2\,\epsilon-3\,\epsilon^2+2\,\epsilon^3)^k$
MAJ_3	$1/4\,(-2\,\epsilon^3+3\,\epsilon^2)^k+1/4\,(1+2\,\epsilon^3-3\,\epsilon^2)^k+3/4\,(2\,\epsilon-3\,\epsilon^2+2\,\epsilon^3)^k+3/4\,(1-2\,\epsilon+3\,\epsilon^2-2\,\epsilon^3)^k$
T_3	$1/8\,(-6\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5+4\,\epsilon^2)^k+1/8\,(1+6\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5-4\,\epsilon^2)^k+1/16\,(\epsilon-\epsilon^2+4\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+1/16\,(1-\epsilon+\epsilon^2-4\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+1/4\,(1-2\,\epsilon+4\,\epsilon^2-6\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+1/4\,(2\,\epsilon-4\,\epsilon^2+6\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+1/8\,(1-\epsilon-\epsilon^2+4\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5)^k+1/8\,(\epsilon+\epsilon^2-4\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5)^k+3/16\,(8\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5+3\,\epsilon-7\,\epsilon^2)^k+3/16\,(1-8\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5-3\,\epsilon+7\,\epsilon^2)^k+3/16\,(2\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5+2\,\epsilon^2+1-2\,\epsilon)^k+3/16\,(-2\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5-2\,\epsilon^2+2\,\epsilon)^k+1/16\,(2\,\epsilon^3-5\,\epsilon^4+2\,\epsilon^5+2\,\epsilon^2)^k+1/16\,(1-2\,\epsilon^3+5\,\epsilon^4-2\,\epsilon^5-2\,\epsilon^2)^k$
T_4	$1/16\,(\epsilon^2+6\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k+1/16\,(1-\epsilon^2-6\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+1/8\,(\epsilon+2\,\epsilon^2-8\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+1/8\,(1-\epsilon-2\,\epsilon^2+8\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k+1/16\,(1-2\,\epsilon+\epsilon^2+6\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k+1/16\,(2\,\epsilon-\epsilon^2-6\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+3/16\,(5\,\epsilon^2-10\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+3/16\,(1-5\,\epsilon^2+10\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k+3/8\,(3\,\epsilon-8\,\epsilon^2+12\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k+3/8\,(1-3\,\epsilon+8\,\epsilon^2-12\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+3/16\,(1-2\,\epsilon+5\,\epsilon^2-10\,\epsilon^3+10\,\epsilon^4-4\,\epsilon^5)^k+3/16\,(2\,\epsilon-5\,\epsilon^2+10\,\epsilon^3-10\,\epsilon^4+4\,\epsilon^5)^k$
MAJ_5	$1/16\,(10\,\epsilon^3-15\,\epsilon^4+6\,\epsilon^5)^k+1/16\,(1-10\,\epsilon^3+15\,\epsilon^4-6\,\epsilon^5)^k+5/16\,(6\,\epsilon^2-14\,\epsilon^3+15\,\epsilon^4-6\,\epsilon^5)^k+5/16\,(1-6\,\epsilon^2+14\,\epsilon^3-15\,\epsilon^4+6\,\epsilon^5)^k+5/8\,(3\,\epsilon-9\,\epsilon^2+16\,\epsilon^3-15\,\epsilon^4+6\,\epsilon^5)^k+5/8\,(1-3\,\epsilon+9\,\epsilon^2-16\,\epsilon^3+15\,\epsilon^4-6\,\epsilon^5)^k$

$\frac{1}{2}$, Theorem 6.9.4 upper-bounds both quantities by $k^{-\nu+o(1)}$. □

6.10 Computer-assisted analysis; $n = 5$

Our problem well avails itself to analysis by computer. In particular, given any explicit function $f\colon \{+1,-1\}^n \to \{+1,-1\}$, a computer mathematics package can easily calculate $\mathcal{P}_k(f;\epsilon)$ exactly as a function of k and ϵ. Furthermore, if n is very small, a computer program can enumerate all antisymmetric left-monotone functions on n bits. We determined there are “only” 135 such functions for $n = 7$ and 2470 such functions for $n = 8$. (The number jumps to 319124 for $n = 9$.) Thus for particular small values of n and k, we can completely solve the problem by comparing an explicit set of polynomials in ϵ on the range $(0, \frac{1}{2})$. As an example, we now analyze the case $n = 5$.

There are exactly 7 antisymmetric left-monotone functions on 5 bits; they are MAJ_1, MAJ_3, MAJ_5, and four functions expressible as thresholds: $T_1 = Th(3,1,1,1,1)$, $T_2 = Th(2,1,1,1,0)$, $T_3 = Th(3,2,2,1,1)$, and $T_4 = Th(4,3,2,2,2)$, where $Th(a_1,\ldots,a_5)$ denotes the function $\mathrm{sgn}(\sum_{i=1}^{5} a_i x_i)$. The table shows $\mathcal{P}_k(f;\epsilon)$ for each of the functions.

For small values of k, we plotted these polynomials for $\epsilon \in (0, \frac{1}{2})$. This led to the following facts, which in principle could be proved by elementary analysis:

Fact 6.10.1

- *For $n = 5$, $2 \le k \le 9$, and any ϵ, the best antisymmetric protocol is* MAJ_1.
- *For $n = 5$, $k = 10, 11$, there exist $0 < \epsilon'_k < \epsilon''_k < \frac{1}{2}$ such that* MAJ_3 *is the best*

antisymmetric protocol for $\epsilon \in [\epsilon'_k, \epsilon''_k]$*, and* MAJ_1 *is the best antisymmetric protocol for all other* ϵ*.*

- *For* $n = 5$*,* $k = 12$*, there exist* $0 < \epsilon'_k < \epsilon''_k < \epsilon'''_k < \frac{1}{2}$ *such that* MAJ_5 *is the best antisymmetric protocol for* $\epsilon \in [\epsilon'_k, \epsilon''_k]$*,* MAJ_3 *is the best antisymmetric protocol for* $\epsilon \in [\epsilon''_k, \epsilon'''_k]$*, and* MAJ_1 *is the best antisymmetric protocol for all other* ϵ*.*

The pattern for $k = 12$ appears to hold for all higher k, with MAJ_5 dominating more and more of the interval, as expected from Theorem 6.7.2.

We will end this section by proving some facts mentioned earlier in this chapter:

Proposition 6.10.2 *There exist* k*,* ϵ*, odd* n*, and odd* $1 < r < n$ *such that* MAJ_r *is a superior protocol to both* MAJ_1 *and* MAJ_n*.*

Proof: Substitute $k = 10$, $\epsilon = .26$ into the table. By explicit calculation, $\mathcal{P}_{10}(\mathsf{MAJ}_1; .26) \leq .0493$, $\mathcal{P}_{10}(\mathsf{MAJ}_5; .26) \leq .0488$, and $\mathcal{P}_{10}(\mathsf{MAJ3}; .26) \geq .0496$. □

Proposition 6.10.3 *There exist* k*,* n*, and a non-majority function* $f \in \mathcal{A}_n$ *such that for each odd* $1 \leq r \leq n$*, there is an* ϵ *for which* $\mathcal{P}_k(f; \epsilon) > \mathcal{P}_k(\mathsf{MAJ}_r; \epsilon)$*.*

Proof: Take $k = 20$, $n = 5$, and $f = Th(3, 1, 1, 1, 1)$. Then by substitution into the table we have $\mathcal{P}_k(f; .01) > .76 > \mathcal{P}_k(\mathsf{MAJ}_3; .01) > \mathcal{P}_k(\mathsf{MAJ}_5; .01)$ and $\mathcal{P}_k(f; .25) > .0034 > \mathcal{P}_k(\mathsf{MAJ}_1; .25)$. □

Proposition 6.10.4 *There exist* n*,* k*, and* $f \in \mathcal{B}_n$ *such that the probability all parties agree on* -1 *differs from the probability all parties agree on* $+1$*.*

Proof: Taking $n = 5$, $k = 3$, and f the left-monotone function whose minterms are $(-1, +1, +1, -1, +1)$ and $(+1, -1, -1, +1, -1)$, explicit calculation gives $\frac{1}{2} - \frac{39}{16}\epsilon + 9\epsilon^2 - \frac{459}{16}\epsilon^3 + \frac{297}{4}\epsilon^4 - \frac{2331}{16}\epsilon^5 + \frac{3465}{16}\epsilon^6 - 234\epsilon^7 + 171\epsilon^8 - 75\epsilon^9 + 15\epsilon^{10}$ and $\frac{1}{2} - \frac{39}{16}\epsilon + \frac{69}{8}\epsilon^2 - \frac{381}{16}\epsilon^3 + \frac{93}{2}\epsilon^4 - \frac{885}{16}\epsilon^5 + \frac{519}{16}\epsilon^6 + 6\epsilon^7 - 24\epsilon^8 + 15\epsilon^9 - 3\epsilon^{10}$ for the probabilities of agreement on -1 and $+1$, respectively. □

Chapter 7

Discussion And Open Problems

We conclude this thesis with a discussion of some results and open problems suggested by our study of noise sensitivity.

7.1 Isoperimetry via the reverse Bonami-Beckner inequality

As mentioned, the reverse Bonami-Beckner inequality does not seem to have been previously used in the study of boolean functions. We believe it should have further interesting applications beyond those in Section 6.9. As an example, we will use it to prove a new isoperimetry or concentration of measure result on the hypercube (c.f. Talagrand [Tal95], Bobkov and Götze [BG99]).

In this section, let us fix $\epsilon \in [0, \frac{1}{2}]$, write $\rho = 1 - 2\epsilon$, and write $\phi_{\Sigma(\rho)}$ for the density function of ρ-correlated bivariate normals, as in Proposition 3.4.1. Explicitly,

$$\begin{aligned}\phi_{\Sigma(\rho)}(x, y) &= (2\pi)^{-1}(1-\rho^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\frac{x^2 - 2\rho xy + y^2}{1-\rho^2}\right) \\ &= (1-\rho^2)^{-\frac{1}{2}}\phi(x)\phi\left(\frac{y - \rho x}{(1-\rho^2)^{\frac{1}{2}}}\right),\end{aligned}$$

where ϕ denotes the standard normal density function.

Theorem 7.1.1 *Let $S, T \subseteq \{+1, -1\}^n$ with $|S| = \exp(-\frac{s^2}{2})2^n$ and $|T| = \exp(-\frac{t^2}{2})2^n$. Then $\mathbf{P}[x \in S, N_\epsilon(x) \in T] \geq 2\pi(1-\rho^2)^{\frac{1}{2}}\phi_{\Sigma(-\rho)}(s, t)$.*

Proof: Take f and g to be the 0-1 characteristic functions of S and T, respectively. Then by Lemma 6.9.3, for any choice of $0 < p, q \leq 1$ with $(1-p)(1-q) = \rho^2$, we get,

$$\mathbf{P}[x \in S, N_\epsilon(x) \in T] \geq \exp(-s^2/2p)\exp(-t^2/2q). \tag{7.1}$$

Write $p = 1 - \rho r$, $q = 1 - \rho/r$ in (7.1), with $r > 0$. Maximizing the right-hand side as a function of r using calculus, the best choice turns out to be $r = \frac{\frac{t}{s}+\rho}{1+\rho\frac{t}{s}}$. (Note that this depends only on the ratio of t to s.) Substituting this choice of r (and hence p and q) into (7.1) yields $\exp(-\frac{1}{2}\frac{s^2+2\rho st+t^2}{1-\rho^2})$, and the proof is complete by definition of $\phi_{\Sigma(-\rho)}$. □

On the other hand, we have the following:

Proposition 7.1.2 *Fix $s, t \geq 1$, and let $S, T \subseteq \{+1, -1\}^n$ be diametrically opposed Hamming balls, with $S = \{x\colon \sum x_i \leq -sn^{\frac{1}{2}}\}$ and $T = \{x\colon \sum x_i \geq tn^{\frac{1}{2}}\}$. Then we have* $\lim_{n\to\infty} \mathbf{P}[x \in S, N_\epsilon(x) \in T] \leq 2\pi(1-\rho^2)^{\frac{1}{2}}\phi_{\Sigma(-\rho)}(s,t)$

Proof: By Proposition 3.4.1, the limit is precisely

$$\int_{-\infty}^{-s}\int_t^\infty \phi_{\Sigma(\rho)}(x,y)\,dy\,dx = \int_s^\infty\int_t^\infty \phi_{\Sigma(-\rho)}(x,y)\,dy\,dx.$$

Let $h(x,y) = \frac{(x+\rho y)(\rho x+y)-\rho(1-\rho^2)}{(1-\rho^2)^2}$. Note that for $x, y \geq 1$, $h(x,y) \geq \frac{(1+\rho)^2-\rho(1-\rho^2)}{(1-\rho^2)^2} = \frac{1+\rho^2}{(1+\rho)(1-\rho)^2} \geq 1$. (Indeed, this inequality holds for a greater range of values for x and y, but we will not try to improve the parameters.) Thus on the range of integration, $\phi_{\Sigma(-\rho)}(x,y) \leq h(x,y)\phi_{\Sigma(-\rho)}(x,y)$. But it may be checked by elementary means that $\int\int h(x,y)\phi_{\Sigma(-\rho)}(x,y)\,dy\,dy = 2\pi(1-\rho^2)^{\frac{1}{2}}\phi_{\Sigma(-\rho)}(x,y)$. The result follows. □

The set S in Proposition 7.1.2 satisfies $\lim_{n\to\infty}|S|2^{-n} = \Phi(-s) \geq \phi(s)(s^{-1}-s^{-3})$ (see [AS72, 26.2.12]), which is quite close to $\exp(-\frac{s^2}{2})$ for large s. A similar statement holds for T. Thus Theorem 7.1.1 is very close to being tight.

As an immediate corollary of Theorem 7.1.1, we have the following:

Corollary 7.1.3 *Let $S \subseteq \{+1,-1\}^n$ have fractional size $\sigma \in [0,1]$, and let $T \subseteq \{+1,-1\}^n$ have fractional size σ^α, for $\alpha \geq 0$. Then if x is chosen uniformly at random from S, then the probability that $N_\epsilon(x)$ is in T is at least $\sigma^{\frac{(\sqrt{\alpha}+\rho)^2}{1-\rho^2}}$. In particular, if $|S| = |T|$,* $\mathbf{P}[N_\epsilon(x) \in T \mid x \in S] \geq \sigma^{\frac{1-\epsilon}{\epsilon}}$.

So, for example, given any two sets with fractional size $\frac{1}{3}$, the probability that applying .3-noise takes a random point chosen from the first set into the second set is at least $(\frac{1}{3})^{.7/.3} \approx 7.7\%$; this probability is nearly achieved by diametrically opposed Hamming balls.

7.2 Open problems

7.2.1 Fourier analysis

Kindler showed that the constant in Theorem 2.4.8 can be taken to be $O(1)/k$; however, this is not known to be tight. If the tribes function is tight for this inequality, then by the calculations at the end of Section 3.9, the correct constant is of a much smaller order: $1/k!$.

Open Question 7.2.1 *Can Theorem 2.4.8 of Talagrand and Benjamini, Kalai, and Schramm be improved to*

$$\sum_{|S|=k} \hat{f}(S)^2 \leq \frac{1+o(1)}{k!} \mathit{II}(f)[\log_2(1/\mathit{II}(f))]^{k-1}$$

for all $1 \leq k \ll \log(1/\mathit{II}(f))$*?*

7.2.2 The reverse Bonami-Beckner inequality

We believe the reverse Bonami-Beckner inequality should have further applications in the study of boolean functions. These applications may be of a somewhat different nature than those of the usual Bonami-Beckner inequality. The reason is that most of the powerful theorems using the usual Bonami-Beckner apply it not to the 0-1 boolean function under study, but to some related function taking on both positive and negative values. In contrast, the reverse Bonami-Beckner inequality only applies to nonnegative functions. As such, it is more likely that it will be useful in direct application to 0-1 boolean functions, as in Theorem 7.1.1.

One area which is a promising candidate for applications of the reverse inequality is that of limitations of distance vs. rate tradeoffs for error correcting codes. The isoperimetric Theorem 7.1.1 already points towards such results; we believe that the reverse inequality may have applications in the Fourier-theoretic study of lower bounds, as in McEliece et al. [MRJW77], Kalai and Linial [KL95] (which describes a failure of the usual Bonami-Beckner inequality in this context), and Linial and Samorodnitsky [LS03].

7.2.3 Noise sensitivity of halfspaces

Perhaps the most interesting tractable problem in analyzing the noise sensitivity of halfspaces is whether our bound for the read-once intersection of halfspaces holds true for general intersections. We conjecture that it does:

Conjecture 7.2.2 *Let f be a function given by the intersection (AND) of k boolean halfspaces over the same set of variables. Then $\mathsf{NS}_\epsilon(f) \leq O(1)(\epsilon \ln k)^{\frac{1}{2}}$.*

Naturally, such a result would be very interesting from a computational learning theory perspective, due to Theorem 5.2.1. Conjecture 7.2.2 may be difficult if it requires intricate analysis of vector-valued random variables of the form $\sum \sigma_i W_i$, where σ_i are i.i.d. ±1 signs and the W_i's are in $\mathbf{R}^n$.

7.2.4 Hardness amplification

An interesting problem in this area would be to see if the direct product theorem, Theorem 4.0.9, can be "derandomized." That is, if we wish to get hardness for the composite function $g \otimes f$, is it necessary for the inputs to copies of f to be independently randomly chosen? Or can they be chosen using fewer input bits — pairwise independently, say, or from a walk on an expander graph? Impagliazzo and Wigderson [IW97], building on the work of Impagliazzo [Imp95], showed that Yao's XOR Lemma holds in such a setting, using many fewer than kn input bits. Their techniques may well extend to Theorem 4.0.9. Such a "derandomization" might be useful in improving Theorem 4.1.2 down to the expected $\frac{1}{2} + n^{-\frac{1}{2}+\gamma}$.

7.2.5 Learning juntas

The major open goal for this problem, and indeed a central problem for uniform-distribution learning theory, is to give a polynomial time agorithm for learning $O(\log n)$-juntas or even $\omega(1)$-juntas. While the problem seems very difficult, perhaps our improving understanding of noise sensitivity and juntas will allows us to make progress. Right now the most serious difficulty is learning k-juntas of the form $\mathsf{PARITY}_{k/2} \oplus g$, where g is any function on the $k/2$ bits not in the parity. Getting an algorithm running in time $n^{\alpha k}$ for $\alpha < \frac{1}{2}$ would be a major breakthrough.

The current bottleneck between $n^{.704k}$ and $n^{\frac{2}{3}k}$ is due to the problem of strongly balanced juntas. A. Kalai has asked the following question:

Open Question 7.2.3 *Is it true that for any boolean function f on k bits which is strongly balanced up to size $\frac{2}{3}k$, there is a restriction fixing at most $\frac{2}{3}k$ bits under which f becomes a parity function?*

If the answer were yes, then it would be straightforward to give a learning algorithm for k-juntas running in time $n^{\frac{2}{3}k}$.

Finally, our junta learning algorithm does not seem to generalize easily to extended versions of the problem. Is it possible to learn juntas under any fixed product distribution besides the uniform one in time $n^{(1-\Omega(1))k}$? Is it possible to learn ternary juntas (i.e. functions on $\{0,1,2\}^n$ with k relevant variables) under the uniform distribution with non-trivial efficiency?

7.2.6 The cosmic coin problem

In the cosmic coin problem, the problem of determining the best protocol for each k, n, and ϵ seems quite delicate; it might not have a simple explicit answer. The main open conjecture regarding this problem was mentioned already in Section 6.1:

Conjecture 6.3.1 *For any k, ϵ, and odd n, there is an odd $1 \leq r \leq n$ such that $\mathcal{P}_k(k;\epsilon)$ is maximized among antisymmetric functions by $f = \mathsf{MAJ}_r$.*

Another very interesting and perhaps easier conjecture is that the majority on a very large number of bits is not worse than the optimal majority by more than a constant factor:

Conjecture 7.2.4 *There is a universal constant $C < \infty$ such that for every k, ϵ,*

$$\mathcal{P}_k(\mathsf{MAJ}_{n^*};\epsilon) \leq C \lim_{\substack{n\to\infty \\ n \text{ odd}}} \mathcal{P}(\mathsf{MAJ}_n, k, \epsilon),$$

where n^* is any odd number (presumably maximizing $\mathcal{P}_k(\mathsf{MAJ}_{n^*};\epsilon)$). That is, the limiting value of $\mathcal{P}_k(\mathsf{MAJ}_n;\epsilon)$ is no worse than the success probability of the best majority, up to a constant factor.

The worst constant C we know to be necessary in Conjecture 7.2.4 is $\frac{\pi}{2}$, from the case $k = 2$, $\epsilon \to \frac{1}{2}$. Resolving both of the above conjectures would give an essentially complete (up to constant factors) solution to the cosmic coins problem for all n, k, and ϵ.

We end with a conjecture strictly weaker than Conjecture 6.3.1 which might be substantially easier to prove:

Conjecture 7.2.5 *For any k, ϵ, and n, the best protocol among all antisymmetric threshold functions is an unweighted antisymmetric threshold function; i.e., a majority.*

Bibliography

[ABC$^+$02] C. Ané, S. Blachère, D. Chafaï, P. Fougères, I. Gentil, F. Malrieu, C. Roberto, and G. Scheffer. *Sur les inégalités de Sobolev logarithmiques.* Société Mathématique de France, 2002.

[An95] M. An. Log-concave probability distributions: Theory and statistical testing. Economics working paper archive at wustl, Washington University at St. Louis, 1995.

[AS72] M. Abramowitz and I. Stegun. *Handbook of mathematical functions.* Dover, 1972.

[AV90] D. Aldous and U. Vazirani. A Markovian extension of Valiant's learning model. In *Proc. 31st Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 392–404, 1990.

[Bak92] D. Bakry. *L'hypercontractivité et son utilisation en théorie des semigroupes.* Springer-Verlag, 1992.

[Ban65] J. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[Bau91a] E. Baum. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Trans. on Neural Networks*, 2:5–19, 1991.

[Bau91b] E. Baum. A polynomial time algorithm that learns two hidden unit nets. *Neural Computation*, 2:510–522, 1991.

[BCGS98] A. Blum, P. Chalasani, S. Goldman, and D. Slonim. Learning with unreliable boundary queries. *Journal of Computing and Sys. Sci.*, 56(2):209–222, 1998.

[Bec75] W. Beckner. Inequalities in Fourier analysis. *Ann. of Math.*, pages 159–182, 1975.

[BEHW87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[Ber98] A. Bernasconi. *Mathematical techniques for the analysis of boolean functions*. PhD thesis, Institute for Computational Mathematics, Pisa, 1998.

[Ber01] A. Bernasconi. On a hierarchy of boolean functions hard to compute in constant depth. *Discrete Math. and Theoretical Comp. Sci.*, 4(2):79–90, 2001.

[Bez94] S. Bezrukov. Isoperimetric problems in discrete spaces. In P. Frankl and Z. Furedi and G. Katona and D. Miklós, editor, *Extremal problems for finite sets*, number 3. Bolyai Soc. Math. Stud., 1994.

[BF02] N. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2:359–395, 2002.

[BFH94] P. Bartlett, P. Fischer, and K.-U. Höffgen. Exploiting random walks for learning. In *Proc. 7th Ann. Workshop on Comp. Learning Theory*, pages 318–327, 1994.

[BFJ$^+$94] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proc. 26th Ann. ACM Symp. on the Theory of Computing*, pages 253–262, 1994.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BG99] S. Bobkov and F. Götze. Discrete isoperimetric and Poincaré-type inequalities. *Prob. Theory and Related Fields*, 114:245–277, 1999.

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability — towards tight results. *SIAM Journal on Computing*, 27:804–915, 1998.

[BJ02] R. Blei and S. Janson. Rademacher chaos: tail estimates vs. limit theorems. To appear, *Arkiv för Matematik*.

[BJT99a] N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC learning of DNF with membership queries under the uniform distribution. In *Proc. 12th Ann. Workshop on Comp. Learning Theory*, pages 286–295, 1999.

[BJT99b] N. Bshouty, J. Jackson, and C. Tamon. Uniform-distribution attribute noise learnability. In *Proc. 12th Ann. Workshop on Comp. Learning Theory*, pages 75–80, 1999.

[BK97a] A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces under a uniform distribution. *Journal of Computing and Sys. Sci.*, 54(2):371–380, 1997.

[BK97b] J. Bourgain and G. Kalai. Influences of variables and threshold intervals under group symmetries. *Geom. and Func. Analysis*, 7:438–461, 1997.

[BKK$^+$92] J. Bourgain, J. Kahn, G. Kalai, Y. Katznelson, and N. Linial. The influence of variables in product spaces. *Israel Journal of Mathematics*, 77:55–64, 1992.

[BKS99] I. Benjamini, G. Kalai, and O. Schramm. Noise sensitivity of boolean functions and applications to percolation. *Inst. Hautes Études Sci. Publ. Math.*, 90:5–43, 1999.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1/2):245–271, 1997.

[Blo62] H. Block. The Perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962.

[Blu94] A. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. In AAAI Fall Symposium on 'Relevance', 1994.

[BMOS03] N. Bshouty, E. Mossel, R. O'Donnell, and R. Servedio. Learning DNF from random walks. To appear, 2003.

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

[Bon68] A. Bonami. Ensembles $\Lambda(p)$ dans le dual de D^∞. *Ann. Inst. Fourier*, 18(2):193–204, 1968.

[Bon70] A. Bonami. Études des coefficients Fourier des fonctiones de $L^p(G)$. *Ann. Inst. Fourier*, 20(2):335–402, 1970.

[Bor82] C. Borell. Positivity improving operators and hypercontractivity. *Math. Z.*, 180(2):225–234, 1982.

[Bou01] J. Bourgain. On the distribution of the Fourier spectrum of boolean functions. To appear in *Israel Journal of Mathematics*.

[Bou99] J. Bourgain. An appendix to *Sharp thresholds of graph properties, and the k-sat problem*, by E. Friedgut. *J. American Math. Soc.*, 12(4):1017–1054, 1999.

[BT96] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

[CG02] H. Chockler and D. Gutfreund. A lower bound for testing juntas. Manuscript, 2002.

[CT68] T. Cover and J. Thomas. *Elements of information theory*. John Wiley & Sons, 1968.

[CW90] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Jounal of Symbolic Comp.*, 9:1–6, 1990.

[Dod00] Y. Dodis. Impossibility of black-box reduction from non-adaptively to adaptively secure coin-flipping. *Electronic Colloq. on Comp. Complexity (ECCC)*, (039), 2000.

[DR02] Y. Ding and M. Rabin. *Hyper-encryption and everlasting security*, volume 2285 of *Lecture Notes in Computer Science*, pages 1–26. 2002.

[DS02] I. Dinur and S. Safra. The importance of being biased. In *Proc. 34th Ann. ACM Symp. on the Theory of Computing*, pages 33–42, 2002.

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

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

[FK98] E. Friedgut and J. Kahn. The number of copies of one hypergraph in another. *Israel Journal of Mathematics*, 105:251–256, 1998.

[FKN01] E. Friedgut, G. Kalai, and A. Naor. Boolean functions whose Fourier transform is concentrated at the first two levels. To appear, *Advances in Mathematics.*

[FKR$^+$02] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. In *Proc. 43rd Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 103–112, 2002.

[Fra87] P. Frankl. *The shifting technique in extremal set theory*, pages 81–110. Cambridge University Press, 1987.

[Fri02] E. Friedgut. Influences in product spaces: KKL and BKKKL revisited. To appear, *Combinatorics, Probability, and Computing.*

[Fri98] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):474–483, 1998.

[Gam99] D. Gamarnik. Extension of the PAC framework to finite and countable Markov chains. In *Proc. 12th Ann. Workshop on Comp. Learning Theory*, pages 308–317, 1999.

[GHM94] M. Golea, T. Hancock, and M. Marchand. On learning μ-perceptron networks on the uniform distribution. *IEEE Trans. on Neural Networks*, 9:67–82, 1994.

[Gre02] B. Green. Lecture notes on Restriction and Kakeya Phenomena — Beckner's inequality. `http://www.dpmms.cam.ac.uk/~bjg23/rkp.html`.

[Gro75] L. Gross. Logarithmic Sobolev inequalities. *Amer. J. of Math.*, 97:1061–1083, 1975.

[Gro92] L. Gross. *Logarithmic Sobolev inequalities and contractivity properties of semigroups.* Springer-Verlag, 1992.

[GTT99] D. Guijarro, J. Tarui, and T. Tsukiji. Finding relevant variables in PAC model with membership queries. In *Proc. 10th Ann. Conference on Algorithmic Learning Theory*, volume 1720, pages 313–322. Springer, 1999.

[Hås97] J. Håstad. Some optimal inapproximability results. In *Proc. 29th Ann. ACM Symp. on the Theory of Computing*, pages 1–10, 1997.

[Hås01] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–869, 2001.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HM91] T. Hancock and Y. Mansour. Learning monotone k-μ DNF formulas on product distributions. In *Proc. 4th Ann. Workshop on Comp. Learning Theory*, pages 179–193, 1991.

[HSW92] D. Helmbold, R. Sloan, and M. Warmuth. Learning integer lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proc. 26th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 538–545, 1995.

[IW97] R. Impagliazzo and A. Wigderson. $\mathsf{P} = \mathsf{BPP}$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proc. 29th Ann. ACM Symp. on the Theory of Computing*, pages 220–229, 1997.

[Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computing and Sys. Sci.*, 55:414–440, 1997.

[JKS02] J. Jackson, A. Klivans, and R. Servedio. Learnability beyond AC^0. In *Proc. 34th Ann. ACM Symp. on the Theory of Computing*, pages 776–784, 2002.

[Kal02] G. Kalai. A Fourier-theoretic perspective on the Concordet paradox and Arrow's theorem. *Adv. in Appl. Math.*, 29(3):412–426, 2002.

[Kea90] M. Kearns. *The computational complexity of machine learning.* The MIT Press, 1990.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[Kha95] M. Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *Journal of Computing and Sys. Sci.*, 50:600–610, 1995.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th Ann. ACM Symp. on the Theory of Computing*, pages 767–775, 2002.

[Kin02] G. Kindler. *Property testing, PCP, and juntas.* PhD thesis, Tel Aviv University, 2002.

[Kin03] G. Kindler. Personal communication, 2003.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proc. 29th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 68–80, 1988.

[KL95] G. Kalai and N. Linial. On the distance distribution of codes. *IEEE Trans. on Info. Theory*, 41:1467–1472, 1995.

[Kle66] D. Kleitman. Families of non-disjoint subsets. *J. Combin. Theory*, 1:153–155, 1966.

[Kli03] C. Klivans. *Combinatorial properties of shifted complexes.* PhD thesis, Massachusetts Institute of Technology, 2003.

[KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[KM02] A. Kalai and Y. Mansour. Personal communication, 2002.

[KOS02] A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. In *Proc. 43rd Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 177–186, 2002.

[KP98] S. Kwek and L. Pitt. PAC learning intersections of halfspaces with membership queries. *Algorithmica*, 22(1/2):53–75, 1998.

[KS02] G. Kindler and S. Safra. Noise-resistant boolean functions are juntas. Manuscript.

[KS01] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proc. 33rd Ann. ACM Symp. on the Theory of Computing*, pages 258–265, 2001.

[KS03] A. Klivans and R. Servedio. Boosting and hard-core sets. *Machine Learning*, 53(2):217–238, 2003.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LS03] N. Linial and A. Samorodnitsky. Linear codes and character sums. *Combinatorica*, 22(4):497–522, 2003.

[Man94] Y. Mansour. *Learning boolean functions via the Fourier transform*, pages 391–424. 1994.

[Man95] Y. Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computing and Sys. Sci.*, 50:543–550, 1995.

[Man01] Y. Mansour. Personal communication, 2001.

[Mau97] U. Maurer. *Information-theoretically secure secret-key agreement by NOT authenticated public discussion*, volume 1233 of *Lecture Notes in Computer Science*, pages 209–225. 1997.

[MO02a] E. Mossel and R. O'Donnell. Coin flipping from a cosmic source: On error correction of truly random bits. To appear, 2002.

[MO02b] E. Mossel and R. O'Donnell. On the noise sensitivity of monotone functions. In B. Chauvin, P. Flajolet, D. Gardy, and A. Mokkadem, editors, *Trends in Mathematics: Mathematics and Computer Science II.* Birkhäuser, 2002.

[MORS03] E. Mossel, R. O'Donnell, O. Regev, and B. Sudakov. Applications of the reverse Bonami-Beckner inequality. Ongoing work, 2003.

[MOS03] E. Mossel, R. O'Donnell, and R. Servedio. Learning juntas. In *Proc. 35th Ann. ACM Symp. on the Theory of Computing*, 2003.

[MRJW77] R. McEliece, E. Rodemich, H. Rumsey Jr. and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams identities. *IEEE Trans. on Info. Theory*, 23:157–166, 1977.

[Nel73] E. Nelson. The free Markov field. *J. of Functional Analysis*, 12:211–227, 1973.

[O'D02] R. O'Donnell. Hardness amplification within NP. In *Proc. 34th Ann. ACM Symp. on the Theory of Computing*, pages 751–760, 2002.

[Per02] Y. Peres. Personal communication, 2002.

[Pet95] V. Petrov. *Limit theorems of probability theory.* Oxford Science Publications, Oxford, England, 1995.

[PRS01] M. Parnas, D. Ron, and A. Samorodnitsky. Proclaiming dictators and juntas, or testing boolean formulae. *Lecture Notes in Computer Science*, 2129:273–284, 2001.

[Raz95] R. Raz. Fourier analysis for probabilistic communication complexity. *Computational Complexity*, 5(3/4):205–221, 1995.

[Reg03] O. Regev. Personal communication, 2003.

[Ros58] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[Rud60] W. Rudin. Trigonometric series with gaps. *J. of Math. and Mech.*, 9:203–227, 1960.

[Saz81] V. Sazonov. *Normal approximation — some recent advances.* Springer-Verlag, 1981.

[Ser01] R. Servedio. *Efficient algorithms in computational learning theory.* PhD thesis, Harvard University, 2001.

[Sha01] R. Shaltiel. Towards proving strong direct product theorems. *Electronic Colloq. on Comp. Complexity (ECCC)*, (009), 2001.

[ST99] O. Schramm and B. Tsirelson. Trees, not cubes: hypercontractivity, cosiness, and noise stability. *Electron. Comm. Probab.*, 4(6):39–49, 1999.

[Šte02] D. Štefankovič. Fourier transforms in computer science. Master's thesis, University of Chicago, 2002.

[STV01] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computing and Sys. Sci.*, 62(2):236–266, 2001.

[Tal94] M. Talagrand. On Russo's approximate 0-1 law. *Annals of Probability*, 22:1476–1387, 1994.

[Tal95] M. Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Inst. Hautes Études Sci. Publ. Math.*, 81:73–205, 1995.

[Tal96] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.

[Tal97] M. Talagrand. On boundaries and influences. *Combinatorica*, 17(2):275–285, 1997.

[Tsi99] B. Tsirelson. Noise sensitivity on continuous products: an answer to an old question of J. Feldman, 1999. math.PR/9907011.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vem97] S. Vempala. A random sampling based algorithm for learning the intersection of halfspaces. In *Proc. 38th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 508–513, 1997.

[Ver90] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proc. 3rd Ann. Workshop on Comp. Learning Theory*, pages 314–326, 1990.

[vzGR97] J. von zur Gathen and J. Roche. Polynomials with two values. *Combinatorica*, 17(3):345–362, 1997.

[Yao82] A. Yao. Theory and application of trapdoor functions. In *Proc. 23rd Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 80–91, 1982.