Electronic Colloquium on Computational Complexity, Report No. 55 (2008)

Some Topics in Analysis of Boolean Functions

Ryan O'Donnell
Carnegie Mellon University
odonnell@cs.cmu.edu

Abstract

This article accompanies a tutorial talk given at the 40th ACM STOC conference. In it, we give a brief introduction to Fourier analysis of boolean functions and then discuss some applications: Arrow's Theorem and other ideas from the theory of Social Choice; the Bonami-Beckner Inequality as an extension of Chernoff/Hoeffding bounds to higher-degree polynomials; and, hardness for approximation algorithms.

1 Introduction

In this article we will discuss boolean functions,

$$f : \{0,1\}^n \to \{0,1\}.$$

Actually, let's agree to write -1 and 1 instead of 0 and 1, so a boolean function looks like

$$f : \{-1,1\}^n \to \{-1,1\}.$$

Boolean functions appear frequently in theoretical computer science and mathematics; they may represent the desired operation of a circuit, the (indicator of) a binary code, a learning theory "concept", a set system over n elements, etc.

Suppose you have a problem (involving boolean functions) with the following two characteristics:

- the Hamming distance, or discrete-cube edge structure on $\{-1,1\}^n$, is relevant;
- you are counting strings, or the uniform probability distribution on $\{-1,1\}^n$ is involved.

These are the hallmarks of a problem for which *analysis of boolean functions* may help. By analysis of boolean functions, roughly speaking we mean deriving information about boolean functions by looking at their "Fourier expansion".

1.1 The "Fourier expansion"

Given a boolean function $f : \{-1,1\}^n \to \{-1,1\}$, interpret the domain $\{-1,1\}^n$ as 2^n points lying in $\mathbb{R}^n$, and think of f as giving a ± 1 labeling to each of these points. There is a familiar method for interpolating such data points with a polynomial. For example, suppose $n = 3$ and f is the "Majority" function Maj_3, so $\mathrm{Maj}_3(1,1,1) = 1$, $\mathrm{Maj}_3(1,1,-1) = 1$, $\ldots$, $\mathrm{Maj}_3(-1,-1,-1) = -1$. Denoting $x = (x_1, x_2, x_3)$, we can write

$$\begin{aligned}\mathrm{Maj}_3(x) &= \left(\tfrac{1+x_1}{2}\right)\left(\tfrac{1+x_2}{2}\right)\left(\tfrac{1+x_3}{2}\right)\cdot(+1)\\ &+ \left(\tfrac{1+x_1}{2}\right)\left(\tfrac{1+x_2}{2}\right)\left(\tfrac{1-x_3}{2}\right)\cdot(+1)\\ &+ \qquad\cdots\\ &+ \left(\tfrac{1-x_1}{2}\right)\left(\tfrac{1-x_2}{2}\right)\left(\tfrac{1-x_3}{2}\right)\cdot(-1).\end{aligned}$$

ISSN 1433-8092

If we actually expand out all of the products, tremendous cancellation occurs and we get

$$\mathrm{Maj}_3(x) = \frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_1x_2x_3. \tag{1}$$

We could do a similar interpolate/expand/simplify procedure even for a function $f : \{-1,1\}^n \to \mathbb{R}$, just by multiplying each x-interpolator by the desired value $f(x)$. And notice that after expanding and simplifying, the resulting polynomial will always be "multilinear" — i.e., have no variables squared, cubed, etc. In general, a multilinear polynomial over variables $x_1, \dots, x_n$ has 2^n terms, one for each monomial $\prod_{i \in S} x_i$, where $S \subseteq [n] := \{1, \dots, n\}$. (Note: $\prod_{i \in \emptyset} x_i$ denotes 1.) Hence:

Proposition 1.1. *Every function $f : \{-1,1\}^n \to \mathbb{R}$ can be uniquely*[1] *expressed as a multilinear polynomial,*

$$f(x) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i, \tag{2}$$

where each c_S is a real number.

This expression (2) is precisely the "Fourier expansion" of f. It is traditional to write the coefficient c_S as $\widehat{f}(S)$ and the monomial $\prod_{i \in S} x_i$ as $\chi_S(x)$; thus we usually see

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S(x). \tag{3}$$

For example, from (1) we can read off the "Fourier coefficients" of the function Maj_3:

$$\begin{gathered}
\widehat{\mathrm{Maj}_3}(\emptyset) = 0, \\
\widehat{\mathrm{Maj}_3}(\{1\}) = \widehat{\mathrm{Maj}_3}(\{2\}) = \widehat{\mathrm{Maj}_3}(\{3\}) = \frac{1}{2}, \\
\widehat{\mathrm{Maj}_3}(\{1,2\}) = \widehat{\mathrm{Maj}_3}(\{1,3\}) = \widehat{\mathrm{Maj}_3}(\{2,3\}) = 0, \\
\widehat{\mathrm{Maj}_3}(\{1,2,3\}) = \frac{1}{2}.
\end{gathered}$$

The "Fourier expansion" gets its name because it can be developed in a more formal way in connection with classical harmonic analysis — with group theory and characters and so on. But it's often just as well to think of it simply as writing $f : \{-1,1\}^n \to \{-1,1\}$ as a polynomial.

1.2 Outline of this article

The aim of this article is to explain some basic concepts in analysis of boolean functions, and then illustrate how they arise is a few diverse areas. Topics, by section number:

§2 Basics of Fourier analysis.

§3 Bias, influences, energy, and noise stability.

§4 Kalai's Fourier-theoretic proof of Arrow's Theorem.

§5 The Hypercontractive/Bonami-Beckner Inequality.

§6 Hardness of approximation via Dictator testing.

Unfortunately, applications in a very large number of areas have to be completely left out, including in learning theory, pseudorandomness, arithmetic combinatorics, random graphs and percolation, communication complexity, coding theory, metric and Banach spaces, ...

[1] We'll see this later.

2 Fourier expansions

2.1 Random strings and Parseval

Let's begin with some basic properties of the Fourier expansion. As mentioned earlier, a hallmark of Fourier analysis is looking at a boolean function's values on strings x chosen from the uniform probability distribution. Throughout this article we'll write random variables in **boldface** and $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ will invariably denote a uniformly random string from $\{-1,1\}^n$. We can think of generating such an $\boldsymbol{x}$ by choosing each bit $\boldsymbol{x}_i$ independently and uniformly from $\{-1,1\}$.

The most basic result in all of Fourier analysis is:

Parseval's Theorem. *For any $f : \{-1,1\}^n \to \mathbb{R}$,*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})^2].$$

Proof. By the Fourier expansion of f,

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})^2] &= \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\left(\sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(\boldsymbol{x})\right)^2\right] \\
&= \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\sum_{S,T\subseteq[n]} \widehat{f}(S)\widehat{f}(T)\chi_S(\boldsymbol{x})\chi_T(\boldsymbol{x})\right] \\
&= \sum_{S,T\subseteq[n]} \widehat{f}(S)\widehat{f}(T) \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\chi_S(\boldsymbol{x})\chi_T(\boldsymbol{x})\right]. \qquad (4)
\end{aligned}$$

Recalling that $\chi_S(x)$ denotes $\prod_{i \in S} x_i$, we see:

Fact 2.1. $\chi_S(x)\chi_T(x) = \chi_{S\triangle T}(x)$.

This is because whenever $i \in S \cap T$ we get an x_i^2, which can be replaced by 1. So we continue:

$$(4) = \sum_{S,T\subseteq[n]} \widehat{f}(S)\widehat{f}(T) \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\chi_{S\triangle T}(\boldsymbol{x})\right]. \qquad (5)$$

We now observe:

Fact 2.2. $\mathbf{E}_{\boldsymbol{x}}[\chi_U(\boldsymbol{x})] = 0$, *unless $U = \emptyset$, in which case it's* 1.

This holds because by independence of the random bits $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ we have

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[\chi_U(\boldsymbol{x})] = \mathop{\mathbf{E}}_{\boldsymbol{x}}[\prod_{i\in U} \boldsymbol{x}_i] = \prod_{i\in U} \mathop{\mathbf{E}}_{\boldsymbol{x}}[\boldsymbol{x}_i],$$

and each $\mathbf{E}[\boldsymbol{x}_i] = 0$. Finally, we deduce

$$(5) = \sum_{S\triangle T=\emptyset} \widehat{f}(S)\widehat{f}(T) = \sum_{S\subseteq[n]} \widehat{f}(S)^2,$$

as claimed. □

(By the way, you can use Parseval to deduce the uniqueness mentioned Proposition 1.1; the proof is an exercise.)

Using linearity of expectation and Facts 2.1 and 2.2, we can easily derive the following formula for the Fourier coefficients of f:

Fact 2.3. *For any $f : \{-1,1\}^n \to \mathbb{R}$ and $S \subseteq [n]$,*

$$\widehat{f}(S) = \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})\chi_S(\boldsymbol{x})].$$

Finally, for ordinary boolean functions $f : \{-1,1\}^n \to \{-1,1\}$ we have $f(x)^2 = 1$ for every x; hence Parseval has the following very important corollary:

Fact 2.4. *If $f : \{-1,1\}^n \to \{-1,1\}$ then*

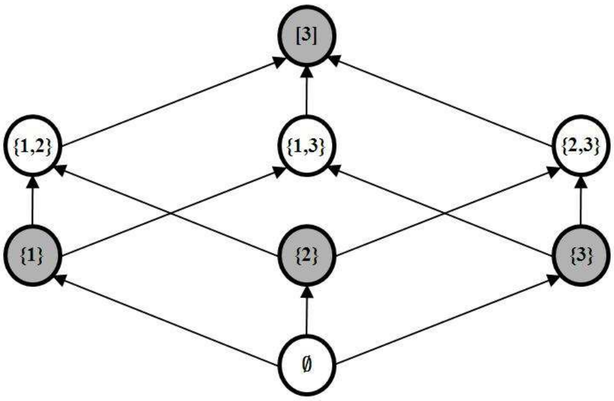
$$\sum_{S\subseteq[n]} \widehat{f}(S)^2 = 1.$$

2.2 Weights

So we have the nice property that a boolean function's squared Fourier coefficients sum to 1. We think of a boolean function as inducing a set of nonnegative "weights" on the subsets $S \subseteq [n]$, where:

Definition 2.5. *The "(Fourier) weight" of f on S is $\widehat{f}(S)^2$.*

By Fact 2.4, the total weight is 1. It can be helpful to try to keep a mental picture of a function's weight distribution on the "poset" of subsets $S \subseteq [n]$. For example, for the Maj_3 function we have the following distribution of weights,



with white circles indicating weight 0 and shaded circles indicating weight 1/4. We also frequently stratify the subsets $S \subseteq [n]$ according to their cardinality:

Definition 2.6. *The "weight of f at level $0 \le k \le n$" is*

$$\mathcal{W}_k(f) := \sum_{\substack{S\subseteq[n]\\|S|=k}} \widehat{f}(S)^2.$$

2.3 Cast of characters

Let's now review some important n-bit boolean functions:

- The two constant functions, $\mathrm{Const}_1(x) = 1$ and $\mathrm{Const}_{-1}(x) = -1$.
- The n Dictator functions, $\mathrm{Dict}_i(x) = x_i$, for $i \in [n]$.

- $\mathrm{Parity}(x) = \chi_{[n]}(x) = x_1 x_2 \cdots x_n$, which is -1 iff an odd number of input bits are -1. More generally, for $S \subseteq [n]$, $\mathrm{Parity}_S(x) = \chi_S(x)$ is the function which is -1 iff an odd number of the input bits in coordinates S are -1.

- The Majority function, $\mathrm{Maj}_n(x) = \mathrm{sgn}(\sum x_i)$, defined for odd n.

- The "Electoral College" function, $\mathrm{EC}^{(51)}(x)$, defined by breaking up the input bits into 51 "states" of size $n/51$, taking Majority on each state, and then taking Majority of the 51 results. (For simplicity we imagine the 50 states and DC have the same population and 1 electoral college vote each.)

- The "Tribes" function [8], which is a fan-in-s OR of disjoint fan-in-t ANDs. The parameters s and t are arranged so that $\mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Tribes}_n(\boldsymbol{x}) = 1] \approx 1/2$; this involves taking $t \approx \log n - \log \ln n$ and $s \approx n/\log n$.

The Fourier expansion of the Constant, Dictator, and Parity_S functions are plain from their definitions above. All of these functions have their Fourier weight concentrated on a single set; for example, $\widehat{\mathrm{Dict}_i}(\{i\}) = 1$, $\widehat{\mathrm{Dict}_i}(S) = 0$ for $S \neq \{i\}$. Indeed, the following is an easy exercise:

Fact 2.7. *If $f : \{-1,1\}^n \to \{-1,1\}$ has $\mathcal{W}_1(f) = 1$ then f is either a Dictator or a negated-Dictator.*

The Majority function plays a central role in the analysis of boolean functions. There is an explicit formula for the Fourier coefficients of Maj_n in terms of binomial coefficients [27], and several elegant formulas giving estimates as $n \to \infty$. We'll mention here just enough so as to give a picture of the Fourier weight distribution for Maj_n. First, another exercise:

Fact 2.8. *If $f : \{-1,1\}^n \to \{-1,1\}$ is "odd", i.e. $f(-x) = -f(x)$ for all x, then $\widehat{f}(S)$ is nonzero only for odd $|S|$.*

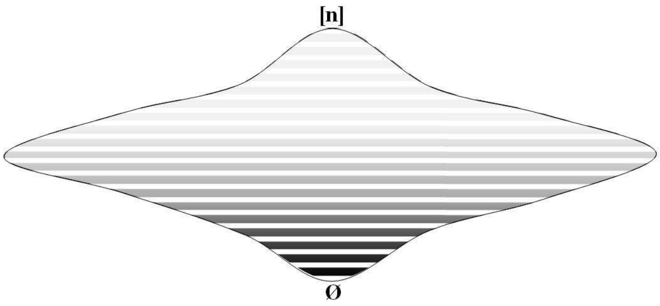
The Majority functions are odd, so they have $\mathcal{W}_k(\mathrm{Maj}_n) = 0$ for all even k. It's also easy to convince yourself that $\widehat{\mathrm{Maj}_n}(S)$ depends only on $|S|$, using the total symmetry of the n coordinates. More interestingly, as we will see in Section 3.2:

Fact 2.9.
$$\lim_{n\to\infty} \mathcal{W}_1(\mathrm{Maj}_n) = 2/\pi.$$

In fact, in several ways Maj_n's Fourier expansion "converges" as $n \to \infty$, so much so that we often speak a little vaguely of "the" Majority function, meaning "Maj_n in the large n limit". For example, we tend to think of Fact 2.9 as saying "Majority has weight $2/\pi$ at level 1". Continuing to speak casually, it holds more generally that for each odd k, $\mathcal{W}_k(\text{“Maj”}) = (2/\pi k)^{3/2} + o(k^{-3/2})$, and hence:

Fact 2.10. *For constant d,*
$$\sum_{k \geq d} \mathcal{W}_k(\text{“Maj”}) = \Theta(1/\sqrt{d}).$$

In other words, we say that Majority has all but ϵ of its Fourier weight below level $O(1/\epsilon^2)$. Actually, the same is true [36] for *any* "weighted majority" function, $f(x) = \mathrm{sgn}(\sum a_i x_i)$, and this fact has played an important role in machine learning — see, e.g., [23]. Below is a weight distribution picture you might keep in mind for "the" Majority function, generalizing the previous picture for Maj_3; darkness corresponds to weight.



For the remaining two functions defined above, Electoral College and Tribes, the Fourier expansion is a little more complicated. For Tribes, explicit formulas for the Fourier coefficients appear in [33]; we'll content ourselves with noting that the weight structure of Tribes is much different from that of Majority: $\mathcal{W}_k(\text{Tribes}_n) = o_n(1)$ for all $0 \leq k \leq n$.

3 Concepts via voting

In this section we explain some interesting quantities associated with a boolean function: bias, influences, energy, and noise stability. Each of these is easily computed from the function's Fourier coefficients. We can motivate these quantities by thinking of $f : \{-1,1\}^n \to \{-1,1\}$ as a *voting rule*. Imagine an election between two parties named -1 and 1. There are n voters, ordered 1, 2, ..., n. We model the ith voter as voting for $\boldsymbol{x}_i \in \{-1,1\}$ uniformly at random, independent of the other voters. (This is the *Impartial Culture Assumption* [14]. It may seem unrealistic, but it is frequently used in the theory of Social Choice. You can think of it as providing a basis for comparing voting rules in the absence of other information.) Finally, we view f as a rule which takes the n votes cast as input, and outputs the winner of the election.

Majority is a popular election rule, but is far from the only possible one. Indeed, looking over the functions from Section 2.3, one can see Electoral College, Dictators, and Constants all occurring "in practice". Even the Tribes function is a vaguely plausible scheme. Only Parity seems unlikely to have ever been used in an actual election. Let's now look at some properties of voting rules $f : \{-1,1\}^n \to \{-1,1\}$, by which we can distinguish them.

3.1 Bias

Definition 3.1. *The "bias" of $f : \{-1,1\}^n \to \{-1,1\}$ is*

$$\mathbf{E}[f] := \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[f(\boldsymbol{x}) = 1] - \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[f(\boldsymbol{x}) = -1].$$

This measures how inherently biased the rule f is in favor of candidate 1 or -1. The connection to Fourier coefficients is immediate from Fact 2.3:

Fact 3.2. $\mathbf{E}[f] = \widehat{f}(\emptyset)$.

The constant functions $\text{Const}_{\pm 1}$ have bias ± 1, whereas Dictators, Majority, Electoral College, and Parity all have bias 0. The bias of Tribes_n is $o_n(1)$. Having zero bias is probably a necessary (but not sufficient) condition for a voting rule f to seem "fair". Losing a little information, we can think of $\widehat{f}(\emptyset)^2 = \mathcal{W}_0(f)$ as measuring the "imbalance" of f, with $\mathcal{W}_0(f) = 0$ meaning f has no bias, and $\mathcal{W}_0(f)$ near 1 meaning f is highly biased.

3.2 Influences

Think of yourself as the ith voter in an election using f. What is the probability your vote makes a difference?

Definition 3.3. *The "influence" of coordinate i on $f : \{-1,1\}^n \to \{-1,1\}$ is*

$$\mathrm{Inf}_i(f) = \Pr_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})],$$

where $\boldsymbol{x}^{\oplus i}$ denotes $\boldsymbol{x}$ with the ith bit negated.

This notion of the "influence" or "power" of a voter was first introduced by Penrose [35]; it was later rediscovered by the lawyer Banzhaf [5] and is usually called the "Banzhaf Power Index" in the Social Choice literature. It has played a role in several United States court decisions [11].

A proof along the lines of Parseval's yields:

Fact 3.4. $\mathrm{Inf}_i(f) = \sum_{S \ni i} \widehat{f}(S)^2$.

In other words, the influence of i on f is equal to the sum of f's weights on sets containing i. Sometimes another formula can be used:

Fact 3.5. *Suppose $f : \{-1,1\}^n \to \{-1,1\}$ is a "monotone" function, meaning $f(x) \geq f(y)$ whenever $x \geq y$ pointwise. Then $\mathrm{Inf}_i(f) = \widehat{f}(\{i\})$.*

Monotonicity is another condition that is probably necessary for a sensible election function f: it means that a vote changing from -1 to 1 can only change the outcome from -1 to 1, and vice versa.

From the definition we can easily see that the n influences on $\mathrm{Const}_{\pm 1}$ are 0, the n influences on the Parity function are 1, and that i is the only coordinate with any influence on Dict_i, having influence 1. You can check that these facts square with Fact 3.4. Also, the fact that $\mathrm{Inf}_i(\mathrm{Parity}) = 1 \neq 0 = \widehat{\mathrm{Parity}}(\{i\})$ shows that the assumption of monotonicity in Fact 3.5 is necessary.

For Maj_n, the ith voter's vote makes a difference if and only if the other $n-1$ votes split exactly evenly. This happens with probability $\binom{n-1}{(n-1)/2} 2^{1-n}$, which by Stirling's formula is asymptotic to $\sqrt{2/\pi}\frac{1}{\sqrt{n}}$. Since Majority is a monotone function we conclude from from Fact 3.5 that $\widehat{\mathrm{Maj}_n}(\{i\})^2 \sim (2/\pi)\frac{1}{n}$ for each i, and thus can derive Fact 2.9.

Finally, for the remaining two functions we've discussed: $\mathrm{Inf}_i(\mathrm{EC}^{(51)}) \approx (2/\pi)\frac{1}{\sqrt{n}}$ for each i, influences slightly smaller than those in Majority; and, $\mathrm{Inf}_i(\mathrm{Tribes}) = \Theta(\frac{\log n}{n})$ for each i, influences much smaller than those in Majority.

3.3 Energy

We now come to the quantity with the most aliases:

Definition 3.6. *The "energy" of $f : \{-1,1\}^n \to \{-1,1\}$ (AKA average sensitivity, total influence, normalized edge boundary, or responsiveness, among other pseudonyms) is*

$$\mathcal{E}(f) = \sum_{i=1}^{n} \mathrm{Inf}_i(f) = \mathop{\mathbf{E}}_{\boldsymbol{x}}[\#\ \textit{of}\ i\ \textit{s.t.}\ f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]. \tag{6}$$

The second equality is just linearity of expectation. From Fact 3.4 we immediately deduce:

Fact 3.7. $\mathcal{E}(f) = \sum_S |S|\widehat{f}(S)^2$.

In other words, the energy of f is the "average" level of its Fourier weight. Having already calculated influences, we get some examples. $\mathcal{E}(\mathrm{Const}_{\pm1}) = 0$ and $\mathcal{E}(\mathrm{Dict}_i) = 1$. Parity is the most energetic function, $\mathcal{E}(\mathrm{Parity}) = n$. Using Fact 3.5 you can show that Majority is the most energetic monotone function, $\mathcal{E}(\mathrm{Maj}_n) \sim \sqrt{2/\pi}\sqrt{n}$. This fact can be incorporated into the mental picture of Majority's Fourier weight distribution from Section 2.3. By contrast, $\mathcal{E}(\mathrm{Tribes}_n) = \Theta(\log n)$; in Section 4.3 we will discuss a sense in which Tribes is the *least* energetic "fair" voting scheme.

Other interpretations of energy follow easily from (6): Thinking of f as a partition of $\{-1,1\}^n$ into two parts, $\mathcal{E}(f)$ is the average number of boundary edges per vertex. Thinking of f as an election rule, $\mathcal{E}(f)$ is the average number of "swing voters". Further, if f is monotone then $\mathcal{E}(f)$ is the expected difference between the votes in favor of the winning candidate and the votes against.

3.4 Noise stability

In a well-run election, the voters' opinions $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are directly fed into f, and $f(\boldsymbol{x})$ is declared the winner. Nothing is perfect, though, and we can conceive that when the ith voter goes to the ballot box, their true vote $\boldsymbol{x}_i$ has a chance of being misrecorded. (Perhaps they tick the wrong box, or the voting machine makes an error.) Denoting the *recorded* votes by $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$, we can ask for the probability that the announced winner, $f(\boldsymbol{y})$, is actually the same as the "true" winner, $f(\boldsymbol{x})$. Of course this depends on our "noise model" for how $\boldsymbol{y}$ is generated from $\boldsymbol{x}$; we'll consider the simplest possible model, where each vote is independently misrecorded with probability ϵ:

Definition 3.8. *Given $0 \le \epsilon \le 1$ we say that $\boldsymbol{x}, \boldsymbol{y} \in \{-1,1\}^n$ are "$(1-2\epsilon)$-correlated" random strings if $\boldsymbol{x}$ is uniformly random and $\boldsymbol{y}$ is generated from $\boldsymbol{x}$ by negating each of its bits independently with probability ϵ.*

The "$1-2\epsilon$" here is because for each i,

$$\mathbf{E}[\boldsymbol{x}_i\boldsymbol{y}_i] = \mathbf{Pr}[\boldsymbol{x}_i = \boldsymbol{y}_i] - \mathbf{Pr}[\boldsymbol{x}_i \neq \boldsymbol{y}_i] = 1 - 2\epsilon.$$

Regarding whether the "true winner wins", we make the following definition:

Definition 3.9. *The "noise stability of f at $1-2\epsilon$" is*

$$\begin{aligned} \mathrm{Stab}_{1-2\epsilon}(f) &= \mathop{\mathbf{E}}_{\substack{\boldsymbol{x},\boldsymbol{y} \\ (1-2\epsilon)\text{-}\textit{correlated}}} [f(\boldsymbol{x})f(\boldsymbol{y})] \\ &= \mathbf{Pr}[f(\boldsymbol{x}) = f(\boldsymbol{y})] - \mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{y})]. \end{aligned}$$

Once again, we can express noise stability in terms of Fourier weights via an easy proof along the line of Parseval's:

Fact 3.10.

$$\begin{aligned} \mathrm{Stab}_{1-2\epsilon}(f) &= \sum_{S\subseteq[n]} (1-2\epsilon)^{|S|}\widehat{f}(S)^2 \\ &= \sum_{k=0}^{n} (1-2\epsilon)^k \mathcal{W}_k(f). \end{aligned}$$

I.e., the noise stability of f at $1-2\epsilon$ is the sum of f's Fourier weights, attenuated by a factor decreasing exponentially with their level. Note also that we can write

$$\mathop{\mathbf{Pr}}_{\substack{\boldsymbol{x},\boldsymbol{y} \\ (1-2\epsilon)\text{-correlated}}} [f(\boldsymbol{x}) = f(\boldsymbol{y})] = \tfrac{1}{2} + \tfrac{1}{2}\mathrm{Stab}_{1-2\epsilon}(f).$$

Let's do some examples. Clearly, $\mathrm{Stab}_{1-2\epsilon}(\mathrm{Const}_{\pm 1}) = 1$; noise in the votes can't possibly change the outcome. For any Dictator we have $\mathrm{Stab}_{1-2\epsilon}(\mathrm{Dict}_i) = 1 - 2\epsilon$; in a dictatorial election, the wrong candidate wins if and only if the Dictator's vote is misrecorded, which happens in our model with probability ϵ. Parity is the next easiest function to consider: using Fact 3.10 and the fact that Parity has all its weight on the set $[n]$, we get that $\mathrm{Stab}_{1-2\epsilon}(\mathrm{Parity}) = (1-2\epsilon)^n$. This is extremely close to 0 assuming $n \gg 1/\epsilon$. In other words, in a (strange!) election where Parity is the election rule, even a little noise in the votes means the announced winner will have almost no correlation with the true winner — they agree with probability only $\frac{1}{2} + \frac{1}{2}(1-2\epsilon)^n$.

The most natural case, when $f = \mathrm{Maj}_n$, is very interesting; we have the following perhaps surprising-looking formula:

Fact 3.11. *For all* $0 \le \epsilon \le 1$*, as* $n \to \infty$*,*

$$\mathrm{Stab}_{1-2\epsilon}(\mathrm{Maj}_n) \to 1 - \tfrac{2}{\pi}\arccos(1-2\epsilon).$$

This fact is well known in the Social Choice literature. The proof has two parts: First, apply the Central Limit Theorem (a 2-dimensional version) to the pair of random variables $\sum \boldsymbol{x}_i$ and $\sum \boldsymbol{y}_i$. Second, use the following formula proved by Sheppard in 1899 [38]: If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are standard Gaussian random variables with $\mathbf{Cov}[\boldsymbol{X}, \boldsymbol{Y}] = 1-2\epsilon$, then $\mathbf{Pr}[\mathrm{sgn}(\boldsymbol{X}) \neq \mathrm{sgn}(\boldsymbol{Y})] = \frac{1}{\pi}\arccos(1-2\epsilon)$. (The algorithmic-minded reader might also recognize this fact from the Goe-mans-Williamson Max-Cut algorithm analysis [16].)

Being a bit less precise, we can use $\arccos(1-2\epsilon) \sim 2\sqrt{\epsilon}$ for small ϵ and hence

$$\mathrm{Stab}_{1-2\epsilon}(\text{“Maj”}) \sim 1 - \tfrac{4}{\pi}\sqrt{\epsilon} \qquad \text{for small } \epsilon.$$

(This fact, combined with Fact 3.10, yields Fact 2.10.) In other words, with majority rule, ϵ-noise in the recording of votes leads to about a $\frac{2}{\pi}\sqrt{\epsilon}$ chance of the wrong winner. With some more probabilistic considerations we can determine that for the Electoral College rule,

$$\mathrm{Stab}_{1-2\epsilon}(\text{“}\mathrm{EC}^{(51)}\text{”}) \sim 1 - 2\left(\tfrac{2}{\pi}\right)^{3/2}\sqrt{51}\sqrt{\epsilon},$$

assuming $51 \ll 1/\epsilon \ll n$. In other words, with the electoral college system there is about a $(2/\pi)^{3/2}\sqrt{51}\sqrt{\epsilon}$ chance that ϵ-noise leads to the wrong winner, higher than that under direct majority by a factor of about 5.7.

For a thorough discussion of the connection between Fourier analysis and Social Choice, see the survey of Kalai [25].

4 Arrow's Theorem, Fair Elections

4.1 Arrow's Theorem

Social Choice theory asks how the preferences of a large population can be aggregated into single choices for society as a whole. This question significantly occupied the Marquis de Condorcet, an eighteenth-century French mathematician and early political scientist. In his 1785 *Essay on the Application of Analysis to the Probability of Majority Decisions* [10] he suggested a method for holding an election between *three* candidates, say A, B, and C. The method is to take the majority preference in each of the pairwise comparisons, A vs. B, B vs. C, and C vs. A, and to use the outcomes as a global ranking of the three candidates. As discussed earlier, we might consider other voting rules besides Majority; given any boolean $f : \{-1,1\}^n \to \{-1,1\}$, let's say the 3-candidate "Condorcet election" works as follows:

	1	2	$\cdots$	n		Society:
A vs. B	$+1$	$+1$	$\cdots$	-1	$=: x$	$f(x)$
B vs. C	-1	$+1$	$\cdots$	$+1$	$=: y$	$f(y)$
C vs. A	$+1$	-1	$\cdots$	$+1$	$=: z$	$f(z)$

Here voter 1's preference is $C > A > B$, voter 2's preference is $A > B > C$, etc. Note that each column (x_i, y_i, z_i) will be one of the six triples

$$\begin{aligned}\{\ &(+1,+1,-1),(+1,-1,+1),(+1,-1,-1),\\ &(-1,+1,+1),(-1,+1,-1),(-1,-1,+1)\ \}.\end{aligned}$$

We call these the "NAE triples", NAE standing for "Not All Equal".

There is a problem with the Condorcet election, as Condorcet himself noted: it can lead to a cycle in the social preference — i.e., the output triple $(f(x), f(y), f(z))$ may not be NAE! In fact, this occurs in the above example if $n = 3$ and $f = \mathrm{Maj}_3$: the output triple is $(+1, +1, +1)$, meaning society seems to rank $A > B > C > A$. This is termed an "irrational outcome", and the fact that it can occur is known as "Condorcet's Paradox". It's not just when $f = \mathrm{Maj}_n$ that this can happen: 165 years later, Arrow famously showed [4] that the Condorcet Paradox can only be avoided in an unappealing way:

Arrow's Impossibility Theorem. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be used for a 3-candidate Condorcet election, and assume f satisfies "unanimity", meaning that $f(1,1,\dots,1) = 1$ and $f(-1,-1,\dots,-1) = -1$. If f is such that the social outcome is never irrational, then f is a Dictator.*

The assumption that a Condorcet election is used is called "independence of irrelevant alternatives" in the usual statement of Arrow's Theorem. The original theorem also allows for using three different aggregating functions f, g, h but it is easy to show that unanimity and no irrationality imply $f = g = h$ (hint: x, $y = -x$, and $z = (f(x), \dots, f(x))$ always consist of NAE input triples, and this implies $g(-x)$ must equal $-f(x)\dots$).

There are very short combinatorial proofs of Arrow's Theorem (see, e.g., [15]). But as we'll see, Gil Kalai's Fourier-analytic proof yields a much more robust conclusion:

Proof. (Kalai [24]) Suppose we use some $f : \{-1,1\}^n \to \{-1,1\}$ (not necessarily satisfying unanimity, even) and ask for the *probability* of an irrational outcome when the n voters' rankings are independent and uniformly random. In other words, the three elections occur with strings $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ where each triple of bits $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)$ is independently chosen to an NAE triple, with probability 1/6 each. Let $\mathrm{NAE} : \{-1,1\}^3 \to \{0,1\}$ denote the indicator function of the NAE triples; then we can write

$$\mathrm{NAE}(a,b,c) = \frac{3}{4} - \frac{1}{4}ab - \frac{1}{4}ac - \frac{1}{4}bc.$$

(This is in fact the Fourier expansion of NAE!) Hence

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[\text{rational outcome}] &= \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[\mathrm{NAE}(f(\boldsymbol{x}), f(\boldsymbol{y}), f(\boldsymbol{z}))]\\ &= \frac{3}{4} - \frac{1}{4}\mathbf{E}[f(\boldsymbol{x})f(\boldsymbol{y})] - \frac{1}{4}\mathbf{E}[f(\boldsymbol{x})f(\boldsymbol{z})] - \frac{1}{4}\mathbf{E}[f(\boldsymbol{y})f(\boldsymbol{z})].\end{aligned}$$

Since $(\boldsymbol{x}, \boldsymbol{z})$ and $(\boldsymbol{y}, \boldsymbol{z})$ have the same joint distribution as $(\boldsymbol{x}, \boldsymbol{y})$, the last three terms are equal; hence the above is

$$\frac{3}{4} - \frac{3}{4}\mathbf{E}[f(\boldsymbol{x})f(\boldsymbol{y})].$$

Now in isolation, what is the distribution on the pair of strings $(\boldsymbol{x}, \boldsymbol{y})$? When $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)$ is a random NAE triple, $\boldsymbol{x}_i$ is uniformly random and $\boldsymbol{y}_i = \boldsymbol{x}_i$ with probability 1/3. Thus $(\boldsymbol{x}, \boldsymbol{y})$ is precisely a pair of $1/3 - 2/3 = (-1/3)$-correlated strings! Hence:

Fact 4.1. *In a 3-candidate Condorcet election,*

$$\begin{aligned}\mathbf{Pr}[f \textit{ yields a rational outcome}] &= \frac{3}{4} - \frac{3}{4}\mathrm{Stab}_{-1/3}(f) \\ &= \frac{3}{4} - \frac{3}{4}\mathcal{W}_0(f) + \frac{1}{4}\mathcal{W}_1(f) - \frac{1}{12}\mathcal{W}_2(f) + \frac{1}{36}\mathcal{W}_3(f) - \cdots\end{aligned}$$

The second equality uses Fact 3.10. Since the sum of all $\mathcal{W}_k(f)$'s is 1, it's clear that if f is to be rational with probability 1 (i.e., if the above quantity is 1), it must be the case that all of f's weight is on level 1. But Fact 2.7 says that $\mathcal{W}_1(f) = 1$ means f is a Dictator or a negated-Dictator, and if f is to have the unanimity property only the former is possible. □

4.2 Robustness of Kalai's proof

Using Fact 4.1 and Fact 3.11 we have:

Fact 4.2. *In a Condorcet election using* Maj_n, *as* $n \to \infty$ *the probability of a rational outcome approaches:*

$$\frac{3}{4} - \frac{3}{4}\left(1 - \frac{2}{\pi}\arccos(-1/3)\right) = \frac{3\arccos(-1/3)}{2\pi} \approx .912.$$

This was first stated in 1952 by Guilbaud [18] and first proved by Garman and Kamien [14]. For brevity, we'll henceforth write simply .912 for "Guilbaud's number", instead of $3\arccos(-1/3)/2\pi$.

So with random voting, Condorcet's Paradox occurs with probability about 8.8% under Majority — small, but not tiny. We might hope that for some other, reasonable non-dictatorial f, the probability of Condorcet's Paradox is negligibly small. The statement of Arrow's Theorem does not rule this out — but Kalai's proof has a robustness which does:

Theorem 4.3. *([25]) Suppose that using* $f : \{-1,1\}^n \to \{-1,1\}$ *in a 3-candidate Condorcet election, the probability of a rational outcome is at least* $1 - \epsilon$. *Then* f *is* $O(\epsilon)$*-close to being a Dictator or a negated-Dictator.*

Here we are using the following definition:

Definition 4.4. *Boolean functions* f *and* g *are "*δ*-close" if* $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \leq \delta$.

To prove Theorem 4.3, Kalai first uses Fact 4.1 to deduce:

Proposition 4.5. *In a 3-candidate Condorcet election using* f, *if the probability of a rational outcome is at least* $1 - \epsilon$, *then* $\mathcal{W}_1(f) \geq 1 - (9/2)\epsilon$.

This is easy to see: if you're limited in how much weight you can put on level 1, your second-best bet is to put it on level 3. To complete the proof, Kalai uses the following "robust" version of Fact 2.7, proven by Friedgut, Kalai, and Naor [13]:

FKN Theorem. *Suppose $f : \{-1,1\}^n \to \{-1,1\}$ satisfies $\mathcal{W}_1(f) \geq 1 - \delta$. Then f is $O(\delta)$-close to being a Dictator or a negated-Dictator. (In fact, $O(\delta)$ can be replaced by $\delta/2 + O(\delta^2)$.)*

4.3 Noise stability and small influences

Theorem 4.3 tells us that we can't hope for any fair election rule that evades Condorcet Paradox with probability close to 1. Given this, we might at least look for the fair election rule that has the highest probability of rational outcomes. To do this, though, we first have to decide what we mean by "fair".

We've already seen a few criteria that seem necessary for an election rule $f : \{-1,1\}^n \to \{-1,1\}$ to be fair: its bias should be 0 and it should be monotone. So far these criteria don't rule out the Dictators, so more is necessary. One way to rule them out would be to require symmetry on on the voters/coordinates. If we insist on total symmetry — i.e., the requirement that $f(\pi(x)) = f(x)$ for every permutation $\pi \in S_n$ — then $f = \mathrm{Maj}_n$ is the only possibility. (Actually, if n is even then there is *no* bias-0, monotone, totally symmetric function.) We can relax this by asking merely for "transitive symmetry". Informally, this is the requirement that "no voter is in a distinguished position"; more formally it means that for every $i, j \in [n]$ there is a permutation π on the coordinates under which f is invariant and which has $\pi(i) = j$. The Electoral College and Tribes functions are transitive symmetric.

It turns out that an even weaker requirement can be used to effectively rule out Dictators — this is the condition of having "small influences".

Definition 4.6. *We say a function $f : \{-1,1\}^n \to \{-1,1\}$ has "τ-small influences" if $\mathrm{Inf}_i(f) \leq \tau$ for all $i \in [n]$.*

Most frequently we informally take τ to be "$o(1)$ with respect to n"; Majority, Electoral College, and Tribes all have $o(1)$-small influences, whereas the Dictators most assuredly do not.

The class of functions with small influences plays an extremely important role in analysis of boolean functions, and many of the more advanced theorems in the field are devoted to properties of this class. Historically, the first such result in the field, due to Kahn, Kalai, and Linial [22], was the following:

KKL Theorem. *No function $f : \{-1,1\}^n \to \{-1,1\}$ with bias 0 has $o\left(\frac{\log n}{n}\right)$-small influences. More generally (cf. Talagrand [39], Friedgut [12]), if f is an unbiased boolean functions with τ-small influences then $\mathcal{E}(f) \geq \Omega(\log(1/\tau))$.*

Since the Tribes_n function has $\mathrm{Inf}_i(\mathrm{Tribes}_n) = \Theta(\frac{\log n}{n})$ for all i, the KKL Theorem is sharp up to constant factors.

Among the 3-candidate Condorcet election rules f with $o(1)$-small influences, which one has the highest probability of rational outcomes? By Fact 4.1 we want the f such that $\mathrm{Stab}_{-1/3}(f)$ is most negative. If f is assumed to be odd (which is also a reasonable requirement for a sensible election rule), Fact 2.8 tells us that $\mathrm{Stab}_{-\rho}(f) = -\mathrm{Stab}_{\rho}(f)$. Hence we might instead look for the odd f with $o(1)$-small influences such that $\mathrm{Stab}_{1/3}(f)$ is largest. Indeed, this problem is interesting for other positive values of $\rho \neq 1/3$, especially ones close to 1: it is the question of finding a "fair" voting rule which is stablest with respect to noise in the recording of votes. To answer these questions, we have the following result [34]:

Majority Is Stablest Theorem. *If $f : \{-1,1\}^n \to \{-1,1\}$ has $o(1)$-small influences,* $\mathbf{E}[f] = 0$, *and* $0 \leq \rho \leq 1$, *then* $\mathrm{Stab}_\rho(f) \leq 1 - \frac{2}{\pi}\arccos(\rho) + o(1)$. *If* $-1 \leq \rho \leq 0$, *then* $\mathrm{Stab}_\rho(f) \geq 1 - \frac{2}{\pi}\arccos(\rho) - o(1)$ *and the assumption* $\mathbf{E}[f] = 0$ *is unnecessary.*

The name of the theorem makes reference to Fact 3.11. We immediately conclude that no f with $o(1)$ influences can be "more rational" than Majority, up to an additive $o(1)$.

It's important in the Majority Is Stablest Theorem that $0 \leq \rho \leq 1$ is first fixed to be a "constant" and that the $o(1)$-smallness of the influences is independent of ρ. In fact, it's not too hard to check that if we fix n and let $\epsilon \to 0$, the quantity $\mathrm{Stab}_{1-2\epsilon}(f)$ approaches $1 - 2\epsilon \cdot \mathcal{E}(f)$. Thus in this regime, maximizing noise stability becomes minimizing energy. This leads us to the question of which function f with $o(1)$-small influences and $\mathbf{E}[f] = 0$ has least energy. The answer (up to constants) is provided by the KKL Theorem: Tribes, with energy $\Theta(\log n)$.

5 Hypercontractivity

Several of the more advanced theorems in the analysis of boolean functions — e.g., the FKN Theorem, the KKL Theorem, and the Majority Is Stablest Theorem — make use of a result called the "Bonami-Beckner Inequality". This inequality was proved first by Bonami [9], proved independently by Gross [17], and then misattributed to Beckner [6] in [22] (Beckner proved generalizations). Due to this confusion of attribution, it might be better to refer to the result by its alternative name, the "Hypercontractive Inequality". The Hypercontractive Inequality has sometimes been described as "deep" or "mysterious"; in this section we hope to demystify it somewhat by explaining it as a generalization of the familiar Hoeffding-Chernoff bounds. A good source for the results appearing in this section is Janson [21].

5.1 The statement

The Hypercontractive Inequality is equivalent to the following (although it is often stated differently):

Hypercontractive Inequality. *Let*

$$f(x) = \sum_{|S| \leq d} \widehat{f}(S)\chi_S(x)$$

denote an arbitrary multilinear polynomial over $x_1, \dots, x_n$ of degree (at most) d. Let $\boldsymbol{F} = f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$, where as usual the $\boldsymbol{x}_i$'s are independent, uniformly random ± 1 bits. Then for all $q \geq p \geq 1$,

$$\|\boldsymbol{F}\|_q \leq \left(\sqrt{\tfrac{q-1}{p-1}}\right)^d \|\boldsymbol{F}\|_p,$$

where $\|\boldsymbol{F}\|_r$ denotes $(\mathbf{E}[|\boldsymbol{F}|^r])^{1/r}$.

It's not hard to prove (say, using Hölder's Inequality) that $\|\boldsymbol{F}\|_r \geq \|\boldsymbol{F}\|_{r'}$ whenever $r \geq r' \geq 1$. The "hyper" in the "Hypercontractive Inequality" refers to the fact that the higher q-norm can be bounded by the lower p-norm — up to a "constant" — if f has low degree.

The special case when $q = 4$ and $p = 2$ is especially useful; in fact, the FKN, KKL, and Majority Is Stablest Theorems only need this case:

Corollary 5.1. *In the setting of the Hypercontractive Inequality,*

$$\mathbf{E}[\boldsymbol{F}^4] \leq 9^d \mathbf{E}[\boldsymbol{F}^2]^2. \tag{7}$$

5.2 Large deviation bounds

One version of the Hoeffding-Chernoff bound says:

Theorem 5.2. *Let $\boldsymbol{F} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \cdots + c_n\boldsymbol{x}_n$, where the $\boldsymbol{x}_i$'s are independent, uniformly random ± 1 bits. Then for all $s \geq 0$,*

$$\mathbf{Pr}[\boldsymbol{F} \geq s] \leq \exp\left(-s^2 \,/\, 2\sum_{i=1}^n c_i^2\right). \tag{8}$$

We can simplify the above statement by writing the parameter s in terms of $\boldsymbol{F}$'s standard deviation. Thinking of $\boldsymbol{F} = f(\boldsymbol{x})$, where $f : \{-1,1\}^n \to \mathbb{R}$ is the linear polynomial $f(x) = c_1x_1 + \cdots + c_nx_n$, Parseval tells us that $\mathbf{E}[f(\boldsymbol{x})^2] = \sum c_i^2$. So

$$\sigma := \operatorname{stddev}(\boldsymbol{F}) = \sqrt{\mathbf{E}[\boldsymbol{F}^2] - \mathbf{E}[\boldsymbol{F}]^2} = \sqrt{\sum c_i^2}, \tag{9}$$

and we can rewrite (8) as

$$\mathbf{Pr}[\boldsymbol{F} \geq t\sigma] \leq \exp(-t^2/2). \tag{10}$$

Thinking of t as large, Hoeffding-Chernoff gives us a very strong upper bound on the probability of a large deviation of $\boldsymbol{F}$ from its mean, 0.

The setup of Theorem 5.2 is just as in the Hypercontractive Inequality with $d = 1$. In this case, Corollary 5.1 tells us that $\mathbf{E}[\boldsymbol{F}^2] \leq 9\mathbf{E}[\boldsymbol{F}^2]^2 = 9\sigma^4$. Actually, it is a nice exercise to check that something slightly better is true:

Proposition 5.3. $\mathbf{E}[\boldsymbol{F}^4] \leq 3\sigma^4 = 3\mathbf{E}[\boldsymbol{F}^2]^2$.

In any case, either bound already gives us a weak version of (10); using Markov's inequality:

$$\mathbf{Pr}[|\boldsymbol{F}| \geq t\sigma] = \mathbf{Pr}[\boldsymbol{F}^4 \geq t^4\sigma^4] \leq \frac{\mathbf{E}[\boldsymbol{F}^4]}{t^4\sigma^4} \leq \frac{3\sigma^4}{t^4\sigma^4} = \frac{3}{t^4}.$$

So we don't get an exponentially small bound, but we still get something polynomially small, better than the $1/t^2$ we'd get from Chebyshev (at least for $t > \sqrt{3}$).

A slightly more complicated (but still elementary) exercise shows that $\mathbf{E}[\boldsymbol{F}^6] \leq 15\sigma^6$, from which the Markov trick yields the large-deviation bound $\mathbf{Pr}[|F| \geq t\sigma] \leq 15/t^6$. This is even better than $3/t^4$, assuming t is large enough. The pattern can be extended for all even integer moments, and qualitatively, being able to bound large moments of $\boldsymbol{F}$ is equivalent to having good "tail bounds" for $\boldsymbol{F}$. (Really, the standard proof of Chernoff's bound works the same way, controlling all moments simultaneously via the Taylor expansion of $\exp(\lambda\boldsymbol{F})$.)

Hypercontractivity gives us similar large-deviation bounds for *higher* degree multilinear polynomials over random ± 1 bits. As in the statement of the Hypercontractive Inequality, let $f(x)$ denote an arbitrary multilinear polynomial over $x_1, \ldots, x_n$ of degree (at most) d, and write $\boldsymbol{F} = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. By subtracting a constant, assume $\mathbf{E}[\boldsymbol{F}] = 0$, in which case

$$\sigma := \operatorname{stddev}(\boldsymbol{F}) = \sqrt{\mathbf{E}[\boldsymbol{F}^2]} = \sqrt{\sum \widehat{f}(S)^2},$$

just as in (9). Now, for example, Corollary 5.1 and Markov's inequality imply that $\mathbf{Pr}[|\boldsymbol{F}| \geq t\sigma] \leq 9^d/t^4$, a most useful result when d is small. Choosing $p = 2$ and $q = q(t)$ carefully, the full Hypercontractive Inequality yields:

Theorem 5.4. *For all $t \geq (2e)^{d/2}$ it holds that*

$$\mathbf{Pr}[|\boldsymbol{F}| \geq t\sigma] \leq \exp\left(-(d/2e) \cdot t^{2/d}\right).$$

5.3 Proof of (4,2)-hypercontractivity

We conclude our discussion of the Hypercontractive Inequality with a simple proof of Corollary 5.1. To the best of our knowledge, this proof first appeared in [34]. Bonami [9] gave a proof somewhat along the same lines for all even integers $q \geq 4$. However, she noted that one has to resort to a more complicated method to prove the Hypercontractive Inequality in its full generality.

Proof. By induction on n (not d!). The base case is $n = 0$. In this case f is just a constant polynomial, $\widehat{f}(\emptyset)$; and even with $d = 0$, both sides of (7) equal $\widehat{f}(\emptyset)^4$.

For the inductive step, we can express the multilinear polynomial f as

$$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_{n-1}) + x_n h(x_1, \ldots, x_{n-1}),$$

where g is a multilinear polynomial of degree at most d and h is a multilinear polynomial of degree at most $d-1$. Introduce the random variables $\boldsymbol{G} = g(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$ and $\boldsymbol{H} = h(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1})$. Now

$$\begin{aligned}
\mathbf{E}[\boldsymbol{F}^4] &= \mathbf{E}[(\boldsymbol{G} + \boldsymbol{x}_n\boldsymbol{H})^4] \\
&= \mathbf{E}[\boldsymbol{G}^4] + 3\mathbf{E}[\boldsymbol{x}_n]\mathbf{E}[\boldsymbol{G}^3\boldsymbol{H}] + 6\mathbf{E}[\boldsymbol{x}_n^2]\mathbf{E}[\boldsymbol{G}^2\boldsymbol{H}^2] \\
&\qquad\qquad + 3\mathbf{E}[\boldsymbol{x}_n^3]\mathbf{E}[\boldsymbol{G}\boldsymbol{H}^3] + \mathbf{E}[\boldsymbol{x}_n^4]\mathbf{E}[\boldsymbol{H}^4] \\
&= \mathbf{E}[\boldsymbol{G}^4] + 6\mathbf{E}[\boldsymbol{G}^2\boldsymbol{H}^2] + \mathbf{E}[\boldsymbol{H}^4],
\end{aligned}$$

where the second step used the fact that $\boldsymbol{x}_n$ is independent of $\boldsymbol{G}$ and $\boldsymbol{H}$. Obviously we should use the induction hypothesis for the first and third terms here; the only "trick" in this proof is to use Cauchy-Schwarz on the middle term. This gives $\mathbf{E}[\boldsymbol{G}^2\boldsymbol{H}^2] \leq \sqrt{\mathbf{E}[\boldsymbol{G}^4]}\sqrt{\mathbf{E}[\boldsymbol{H}^4]}$ and we can now use the induction hypothesis four times, and continue the above:

$$\begin{aligned}
&\leq 9^d\mathbf{E}[\boldsymbol{G}^2]^2 + 6\sqrt{9^d\mathbf{E}[\boldsymbol{G}^2]^2}\sqrt{9^{d-1}\mathbf{E}[\boldsymbol{H}^2]^2} + 9^{d-1}\mathbf{E}[\boldsymbol{H}^2]^2 \\
&= 9^d\left(\mathbf{E}[\boldsymbol{G}^2]^2 + 2\mathbf{E}[\boldsymbol{G}^2]\mathbf{E}[\boldsymbol{H}^2] + \tfrac{1}{9}\mathbf{E}[\boldsymbol{H}^2]^2\right) \\
&\leq 9^d\left(\mathbf{E}[\boldsymbol{G}^2] + \mathbf{E}[\boldsymbol{H}^2]\right)^2.
\end{aligned}$$

But this last quantity is precisely $9^d\mathbf{E}[\boldsymbol{F}^2]^2$, since

$$\begin{aligned}
\mathbf{E}[\boldsymbol{F}^2] &= \mathbf{E}[(\boldsymbol{G} + \boldsymbol{x}_n\boldsymbol{H})^2] \\
&= \mathbf{E}[\boldsymbol{G}^2] + 2\mathbf{E}[\boldsymbol{x}_n]\mathbf{E}[\boldsymbol{G}\boldsymbol{H}] + \mathbf{E}[\boldsymbol{x}_n^2]\mathbf{E}[\boldsymbol{H}^2] \\
&= \mathbf{E}[\boldsymbol{G}^2] + \mathbf{E}[\boldsymbol{H}^2],
\end{aligned}$$

and this completes the induction. □

6 Hardness of approximation

6.1 Overview

Analysis of boolean functions is a very powerful tool for constructing hard instances for approximation algorithms, and for proving NP-hardness of approximation. The canonical example of this occurs in Håstad's $1-\delta$ vs. $1/2+\delta$ NP-hardness result for Max-3Lin [20].[2] Max-3Lin is the constraint satisfaction problem (CSP) over boolean variables $v_1, \dots, v_n$ in which the constraints are of the form $v_i v_j v_k = \pm 1$. (The "Lin" part of the name comes from thinking of boolean values as 0 and 1 mod 2, in which case the constraints are linear: $v_i + v_j + v_k = 0/1 \bmod 2$.) The "optimum value" of a Max-3Lin instance is the fraction of constraints satisfied by the best assignment. A "c vs. s" NP-hardness result for Max-3Lin means that there is no polynomial-time algorithm which can distinguish instances with optimum value at least c from instances with optimum value less than s — unless P = NP. In particular, there can be no s/c-factor approximation algorithm. It's very easy to see that Håstad's $1-\delta$ vs. $1/2+\delta$ NP-hardness is best possible, in that the gap can't be widened to $c = 1$ or $s = 1/2$.

Since Håstad's result, the methodology for proving such strong "inapproximability" results has become almost standardized. We will discuss it an extremely high level in Section 6.4. Briefly, the key to proving inapproximability for a certain CSP is to design a "gadget instance" of that CSP with appropriate properties. Further, for certain reasons having to do with locally testable codes [7], these gadget instances are invariably based on "Dictator vs. Small-influence tests" — a topic tailor-made for the analysis of boolean functions.

6.2 Dictator vs. Small-influences tests

A Dictator vs. Small-influences test is a highly specific kind of Property Testing algorithm. The object to be tested is an unknown boolean function $f : \{-1,1\}^n \to \{-1,1\}$. The property to be tested is that of being one of the n Dictator functions. Finally, there is a severe restriction on the number of queries: we usually want just 2 or 3, and they must be "nonadaptive". To compensate for this, we relax the goal even more than is usual in Property Testing: we only need to reject with high probability the functions that are "very non-dictatorial". Specifically, we use the small-influences criterion discussed in Section 4.3.

Definition 6.1. *A q-query "Dictator vs. Small-influences test" using the predicate $\phi : \{-1,1\}^q \to \{pass, fail\}$ consists of a randomized procedure for choosing strings $\boldsymbol{x}_1, \dots, \boldsymbol{x}_q \in \{-1,1\}^n$. The probability that a function $f : \{-1,1\}^n \to \{-1,1\}$ "passes the test" is $\mathbf{Pr}[\phi(f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_q)) = pass]$. We say the test has "completeness" c if the n dictator functions pass with probability at least c. We say the test has "soundness" s if all f having $o(1)$-small influences[3] pass with probability at most $s + o(1)$. We then say that the test is a "c vs. s Dictator vs. Small-influences test".*

Let's see some examples of Dictator vs. Small-influences tests. Our first example might already be obvious, given the discussion in Section 4. Since we are looking for a way to distinguish dictatorships from non-dictatorships with, say, 3 queries/applications of f, Arrow's Theorem springs immediately to mind. Specifically, the 3-candidate Condorcet election gives such a distinguisher; we call it the "NAE Test":

[2] Throughout this section, δ denotes a positive constant that can be made arbitrarily small.

[3] The $o(1)$ here is with respect to n; we are being slightly informal so as to eliminate additional quantifiers and parameters.

NAE Test: Simulate a random 3-candidate Condorcet election with f (as in Kalai's proof of Arrow's Theorem). Pass/fail f using the predicate $\phi = \text{NAE}$.

If f is a Dictator function then the NAE Test passes with probability 1; the Majority Is Stablest Theorem implies that if f has $o(1)$-small influences it passes the NAE Test with probability at most $.912 + o(1)$. Hence the NAE Test is a 1 vs. .912 Dictator vs. Small-Influences test using the predicate NAE.

Our second example is quite similar; it is a 2-query test using the predicate $\phi =$ "$\neq$".

Noise Stability Test: Pick $\boldsymbol{x}$ and $\boldsymbol{y}$ to be $(-1+2\epsilon)$-correlated strings and pass f iff $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$.

Here $0 \leq \epsilon \leq 1/2$ is thought of as smallish, so $\boldsymbol{x}$ and $\boldsymbol{y}$ are very "anti-correlated". By definition, the probability some f passes this test is $\frac{1}{2} - \frac{1}{2}\text{Stab}_{-1+2\epsilon}(f)$. If f is a Dictator then this probability is $1-\epsilon$. Again, from the Majority Is Stablest Theorem it follows that if f has $o(1)$-small influences, it passes with probability at most

$$\arccos(-1+2\epsilon)/\pi + o(1) = 1 - \arccos(1-2\epsilon)/\pi + o(1).$$

Hence this is a $1-\epsilon$ vs. $1 - \arccos(1-2\epsilon)/\pi$ Dictator vs. Small-influences test using the predicate $\phi =$ "$\neq$".

Finally, Håstad's inapproximability result for Max-3Lin is based on the following:

Håstad's Test: Pick $\boldsymbol{x}, \boldsymbol{y} \in \{-1,1\}^n$ uniformly and independently and let $\boldsymbol{z}$ be a random string which is $(1-2\delta)$-correlated to the string $\boldsymbol{x} \circ \boldsymbol{y}$. (Here $\boldsymbol{x} \circ \boldsymbol{y}$ is the string whose ith coordinate is $\boldsymbol{x}_i \boldsymbol{y}_i$.) Also pick $\boldsymbol{b} \in \{-1,1\}$ uniformly. Pass f iff $f(\boldsymbol{x})f(\boldsymbol{y})f(\boldsymbol{bz}) = \boldsymbol{b}$.

It's easy to check that Dictators pass this test with probability $1-\delta$. More generally, a proof along the lines of Parseval's yields that

$$\begin{aligned} & \mathbf{Pr}[f \text{passes Håstad's test}] \\ = \;& \tfrac{1}{2} + \tfrac{1}{2} \sum_{|S| \text{ odd}} (1-2\delta)^{|S|} \widehat{f}(S)^3 \\ \leq \;& \tfrac{1}{2} + \tfrac{1}{2} \max_{|S| \text{ odd}} \{(1-2\delta)^{|S|} |\widehat{f}(S)|\} \cdot \sum_{S \subseteq [n]} \widehat{f}(S)^2 \\ = \;& \tfrac{1}{2} + \tfrac{1}{2} \max_{|S| \text{ odd}} \{(1-2\delta)^{|S|} |\widehat{f}(S)|\}. \end{aligned}$$

If f has $o(1)$-small influences then $|\widehat{f}(S)|$ must be $o(1)$ for all $S \neq \emptyset$, in particular for all S with $|S|$ odd — this is because of Fact 3.4. Hence every $o(1)$-small influences f passes Håstad's test with probability at most $1/2 + o(1)$. In other words, Håstad's Test is a $1-\delta$ vs. $1/2$ Dictator vs. Small-influences test using the two 3-bit predicates "$v_i v_j v_k = 1$" and "$v_i v_j v_k = -1$".

(You might wonder why we don't just take $\delta = 0$. The reason is that for proving NP-hardness-of-approximation results we technically need something slightly stronger than Dictator vs. Small-influences tests. Specifically, we also need the tests to fail functions which merely have $o(1)$ "low-degree influences"; roughly speaking, those f for which $\sum_{S \ni i} (1-o(1))^{|S|} \widehat{f}(S)^2$ is $o(1)$ for each i.)

6.3 Hard instances

Before describing how Dictator vs. Small-influences tests fit into NP-hardness reductions, it's useful to see how they can at least be viewed as "hard instances" for algorithmic problems. By a hard instance we mean one for which a fixed, standard algorithm — usually LP rounding or SDP rounding — finds a significantly suboptimal solution. (Note that this is different from an "LP/SDP integrality gap" instance.)

Let's see this for one of our Dictator vs. Small-influences tests — say the NAE Test. In preparation for our discussion of inapproximability, we'll talk about testing "K"-bit functions rather than "n"-bit functions. If we were to explicitly write down all possible checks the NAE Test does for a function $f : \{-1,1\}^K \to \{-1,1\}$, it would look like this:

check:	w.p.:
NAE($f(1,1,\dots,1), f(1,1,\dots,1), f(-1,-1,\dots,-1)$)	$1/6^K$
NAE($f(1,1,\dots,1), f(1,1,\dots,-1), f(-1,-1,\dots,1)$)	$1/6^K$
NAE($f(1,1,\dots,1), f(1,1,\dots,-1), f(-1,-1,\dots,-1)$)	$1/6^K$
$\dots$	$\dots$

Now imagine you are an algorithm trying to assign the 2^K values of f so as to maximize the probability it passes this test. Stepping back for a moment, what you are faced with is an *instance* of the CSP over boolean variables in which each constraint is a 3-variable NAE predicate. This CSP is variously called "Max-3NAE-Sat (with no negations)", "Max-3-Set-Splitting", or "2-coloring a 3-uniform hypergraph". We'll call it simply Max-NAE; not the most famous algorithmic problem, perhaps, but a well-studied one nevertheless [26, 1, 41, 42, 19].

Why is the particular instance given by the NAE Test a "hard instance" of Max-NAE? On one hand, its optimal value is 1. (Indeed it has $2K$ optimal solutions, the Dictators and the negated-Dictators.) On the other hand, you can show that for this instance, the best known approximation algorithm (semidefinite programming [42]) may give a solution whose expected value is only .912, Guilbaud's number.

The reason for this is that the SDP may find the optimal feasible vector solution in which each "variable" $f(x)$ gets mapped to the K-dimensional unit vector $x/\sqrt{K}$. In this case the rounding algorithm will output the solution $f(x) = \mathrm{sgn}(\sum \boldsymbol{g}_i x_i)$, where the $\boldsymbol{g}_i$'s are independent Gaussian random variables — a random "weighted majority" function. With high probability over the $\boldsymbol{g}_i$'s, this function will have $o(1)$-small influences and thus pass the NAE Test with probability at most $.912 + o(1)$. In other words, as a solution to the Max-NAE instance, it will have value at most $.912 + o(1)$.

Actually, this is the worst possible such instance for Max-NAE: analysis of Zwick [41] shows that the SDP algorithm achieves value at *least* .912 on Max-NAE instances with optimal value 1.

6.4 NP-hardness results

Having seen that Dictator vs. Small-influences tests can provide hard instances for CSPs, we'll briefly explain how inapproximability technology can turn these hard instances into actual NP-hardness results. For more detailed explanations, see the surveys of Trevisan and Khot [40, 29].

Most strong NP-hardness-of-approximation results are reductions from a problem known as "Label-Cover with K Labels". This problem is known to be NP-hard even to slightly approximate, thanks to the PCP Theorem [3, 2] and the Parallel Repetition Theorem [37]. To reduce from Label-Cover to c vs. s hardness of the CSP "Max-ϕ", you use "gadgets" to encode the K different "labels". These gadgets should be instances of Max-ϕ with the following two properties (vaguely stated): one, they should have K distinct "pure solutions" of value at least c; two, any "generic mixture" of these pure solutions should have value at most s. These are exactly the properties that Dictator vs. Small-influences tests have, when viewed as "hard instances": the Dictators are the pure solutions and the small-influence functions are the generic mixtures.

It is beyond the scope of this article to go into more details; we will content ourselves with the following two statements. First, in *some cases* a c vs. s Dictator vs. Small-influences test using the predicate ϕ can be used as a gadget in a reduction from Label-Cover to conclude c vs. $s + \delta$ NP-hardness-of-approximation for Max-ϕ. This is exactly how Håstad's $1 - \delta$ vs. $1/2 + \delta$ NP-hardness for Max-3Lin works, using his Dictator vs. Small-influences test. Whether or not a Dictator test can be plugged into Label-Cover to get the equivalent hardness result depends in a technical way on the properties of the test.

In *all cases* (modulo minor technical details; see [30]), a c vs. s Dictator vs. Small-influences test using the predicate ϕ can be used as a "Unique-Label-Cover" gadget to conclude $c - \delta$ vs. $s + \delta$ hardness-of-approximation for Max-ϕ. However the "Unique-Label-Cover" problem is not known to actually have the same extreme inapproximability as "Label-Cover" — this is the content of Khot's notorious "Unique Games Conjecture (UGC)" [28]. Thus we can only get NP-hardness results in this way assuming the unproven UGC.

As examples, assuming UGC, the NAE Test gives $1 - \delta$ vs. $.912 + \delta$ hardness for Max-NAE, essentially matching the approximation algorithm of Zwick [41]. (Getting 1 vs. $.912 + \delta$ hardness subject to a UGC-like conjecture is an interesting open problem.) Similarly, assuming UGC, the Noise Stability Test yields $1 - \epsilon - \delta$ vs. $1 - \arccos(1 - 2\epsilon)/\pi + \delta$ hardness for the "Max-$\neq$" problem — i.e., Max-Cut — for any $0 < \epsilon < 1/2$. Taking $\epsilon \approx .155$ we get .845 vs. .742 hardness, for a ratio of .878 which matches the approximation guarantee of the Goemans-Williamson algorithm [16]. Taking ϵ very small, we get the reduction forming the basis of Khot and Vishnoi's unconditional (UGC-less) proof that negative-type metrics do not embed into ℓ_1 with constant distortion [31]. Krauthgamer and Rabani's quantitative improvement of this result [32] used the related fact that Tribes is the unbiased small-influences function with least energy.

The reason the reliance on UGC is becoming widespread is the fact that it converts Dictator vs. Small-influences tests into NP-hardness results in a "black-box" fashion. This reduces the whole question of inapproximability into Property Testing problems, a domain where analysis of boolean functions plays a central role. Perhaps analysis of boolean functions will play a key role in the resolution of the Unique Games Conjecture itself.

7 Acknowledgment

I'd like to thank Cynthia Dwork and the organizers of STOC 2008 for giving me the opportunity to present this tutorial.

References

[1] G. Andersson and L. Engebretsen. Better approximation algorithms for Set Splitting Not-All-Equal-SAT. *Inf. Proc. Lett.*, 65(6):305–311, 1998.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[4] K. Arrow. A difficulty in the concept of social welfare. *J. of Political Economy*, 58(4):328–346, 1950.

[5] J. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[6] W. Beckner. Inequalities in Fourier analysis. *Ann. Math.*, 102(1):159–182, 1975.

[7] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability — towards tight results. *SICOMP*, 27(3):804–915, 1998.

[8] M. Ben-Or and N. Linial. Collective coin flipping, robust voting schemes and minima of Banzhaf values. In *Proc. 26th FOCS*, pages 408–416, 1985.

[9] A. Bonami. Étude des coefficients de Fourier des fonctions de $L^p(G)$. *Ann. Inst. Fourier*, 20(2):335–402, 1970.

[10] N. de Condorcet. Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. *Imprimerie Royale, Paris*, 1785.

[11] D. Felsenthal and M. Machover. *The Measurement of Voting Power: Theory and Practice, Problems and Paradoxes.* Edward Elgar, 1998.

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

[13] E. Friedgut, G. Kalai, and A. Naor. Boolean functions whose Fourier transform is concentrated on the first two levels. *Adv. in Appl. Math*, 29(3):427–437, 2002.

[14] M. Garman and M. Kamien. The paradox of voting: probability calculations. *Behavioral Science*, 13(4):306–16, 1968.

[15] J. Geanakoplos. Three brief proofs of Arrow's Impossibility Theorem. *Economic Theory*, 26(1):211–215, 2005.

[16] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.

[17] L. Gross. Logarithmic Sobolev inequalities. *Amer. J. Math.*, 97:1061–1083, 1975.

[18] G. Guilbaud. Les théories de l'intérêt général et le problème logique de l'agrégration. *Economie appliquée*, 5:501–584, 1952.

[19] V. Guruswami. Inapproximability results for Set Splitting and satisfiability problems with no mixed clauses. *Algorithmica*, 38(3):451–469, 2003.

[20] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[21] S. Janson. *Gaussian Hilbert Spaces*, volume 129 of *Cambridge Tracts in Mathematics.* Cambridge University Press, 1997.

[22] J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *Proc. 29th FOCS*, pages 68–80, 1988.

[23] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proc. 46th FOCS*, pages 11–20, 2005.

[24] G. Kalai. A Fourier-theoretic perspective on the Concordet paradox and Arrow's theorem. *Adv. in Appl. Math.*, 29(3):412–426, 2002.

[25] G. Kalai. Noise sensitivity and chaos in social choice theory. Discussion Paper Series dp399, Center for Rationality and Interactive Decision Theory, Hebrew University, 2005.

[26] V. Kann, J. Lagergren, and A. Panconesi. Approximability of maximum splitting of k-sets and some other APX-complete problems. *Inf. Proc. Lett.*, 58(3):105–110, 1996.

[27] M. Karpovsky. *Finite orthogonal series in the design of digital devices.* John Wiley, 1976.

[28] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th STOC*, pages 767–775, 2002.

[29] S. Khot. Inapproximability results via Long Code based PCPs. *SIGACT News*, 36(2):25–42, 2005.

[30] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SICOMP*, 37(1):319–357, 2007.

[31] S. Khot and N. Vishnoi. The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proc. 46th FOCS*, pages 53–62, 2005.

[32] R. Krauthgamer and Y. Rabani. Improved lower bounds for embeddings into l_1. In *Proc. 17th SODA*, pages 1010–1017, 2006.

[33] Y. Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *J. Comput. Sys. Sci.*, 50(3):543–550, 1995.

[34] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *Proc. 46th FOCS*, pages 21–30, 2005. To appear, *Ann. Math.*

[35] L. Penrose. The elementary statistics of majority voting. *J. of the Royal Statistical Society*, 109(1):53–57, 1946.

[36] Y. Peres. Noise stability of weighted majority. `arXiv:math/0412377v1`, 2004.

[37] R. Raz. A parallel repetition theorem. *SICOMP*, 27(3):763–803, 1998.

[38] W. Sheppard. On the application of the theory of error to cases of normal distribution and normal correlation. *Phil. Trans. Royal Soc. London, Series A*, 192:101–531, 1899.

[39] M. Talagrand. On Russo's approximate zero-one law. *Ann. Prob.*, 22(3):1576–1587, 1994.

[40] L. Trevisan. Inapproximability of combinatorial optimization problems. *Electronic Colloq. on Comp. Complexity (ECCC)*, 065, 2004.

[41] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proc. 9th SODA*, pages 201–210, 1998.

[42] U. Zwick. Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. 31st STOC*, pages 679–687, 1999.