The Harmonic Analysis of Boolean Functions

Jacob Denson

February 17, 2017

Table Of Contents

Chapter 1

Introduction to Boolean Analysis

1.1 Introduction

This course is about harmonic analysis over the domain $\mathbf{F}_2^n$, and the surprising applications of the field to computing science. It seems that whenever one wants to deal with a combinatorial problem dealing with the Hamming distance of two strings $x, y \in \mathbf{F}_2^n$, defined by

$$\Delta(x,y) = \#\{i : x^i \neq y^i\}$$

then Fourier analytic tools tend to come into play. Since any discrete structure can be encoded by binary inputs, where each bit represents a property of that discrete object, then these methods extend to attack problems on any discrete classes of mathematical objects which attempt to quantify properties of these objects in terms of small discrete differences. The methods are particularly applicable when these differences can be directly stated in terms of the binary representation of these objects.

The direct relation of $\mathbf{F}_2$ to computing science should be apparent if you've ever learned a modern programming language, where we see the correspondence between $\mathbf{F}_2$ and the Boolean truth values $\top$ and $\bot$. The group structure on $\mathbf{F}_2$ induces a group structure on truth values, where addition corresponds to the exclusive-or operation

$$A \oplus B = \begin{cases} \top & A \text{ is } \top, \text{ or } B \text{ is } \top, \text{ but not both.} \\ \bot & \text{otherwise} \end{cases}$$

and multiplication corresponds to conjunction. Already we see the hamming distance come into play, because $A \oplus B = \top$ if and only if $A \neq B$. We shall also consider a less obvious, but incredibly important correspondence between the additive structure of $\mathbf{F}_2$ and the multiplicative group $\{-1, 1\}$, mapping 0 to 1, and 1 to -1. Initially, this is a very strange correspondence to pick, because we normally think of 1 as being the value of 'truth'. However, the correspondence does give an isomorphism between the two group structures, and fits with the general convention of performing harmonic analysis over the multiplicative group $\mathbf{T}^n$ where possible. Furthermore, it corresponds with the convention of performing harmonic analysis over subdomains of $\mathbf{T}$. If we wish to consider the Fourier transform of some $f : \mathbf{F}_2^n \to \mathbf{F}_2$, we require a representation of $\mathbf{F}_2$ in $\mathbf{T}$, and the only correspondence possible is with $\{-1, 1\}$.

The characters on the group $\mathbf{T}^n$ take the form of integer-valued monomials $z_1^{m_1} \dots z_n^{m_n}$. Because of the canonical embedding of $\{-1, 1\}^n$ in $\mathbf{T}^n$, every character on $\{-1, 1\}^n$ is the restriction of a character on $\mathbf{T}^n$, and can therefore be represented by a monomial. Since $x_i^2 = 1$ for all $x_i \in \mathbf{F}_2$, these monomials can be chosen with $m_i \in \{0, 1\}$. Abstract harmonic analysis tells us that there are exactly 2^n characters on the group $\mathbf{F}_2^n$, so these are all the identifications we can make, and therefore the functions

$$x^S = \prod_{i \in S} x^i$$

form the set of all characters on $\mathbf{F}_2^n$. Using functional notation and switching domains, we define the characters $\chi_S : \mathbf{F}_2^n \to \{-1, 1\}$ by

$$\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$$

The basic theory of discrete abelian harmonic analysis tells us that the characters x^S form an orthonormal basis for the Hilbert space of real-valued functions on $\mathbf{F}_2^n$, so for every function $f : \{-1, 1\}^n \to \mathbf{R}$ we have a unique expansion

$$f(x) = \sum_{S \subset [n]} \widehat{f}(S) x^S$$

where $\widehat{f}(S) \in \mathbf{R}$ is the *Walsh-Fourier coefficient* corresponding to S. If we consider some index set other than n, that is, if we consider the Fourier transforms of functions on $\mathbf{F}_2^I$ for some arbitrary index set I, then we will

obtain an expansion of the form

$$f(x) = \sum_{S \subset I} \widehat{f}(S) x^S$$

We will occassionally use this notation when it is more natural to do so.

An easy way to find these coeffients, without any harmonic analysis, is to expand the any function $f : \{-1,1\}^n \to \{-1,1\}$ in its conjunctive normal form[1], calculating

$$f(x) = \sum_y \mathbf{I}(x = y) f(y) = \frac{1}{2^n} \sum_y \left(\prod_{i=1}^n 1 + x_i y_i \right) f(y)$$

Using certain algebraic computations, we can use this formula to find a *particular* expansion of a function, and we now know that it is the *unique* expansion. This is essentially a binary equivalent to the technique used to find power series of holomorphic functions, combining well know expansions to form the expansion of a particular formula.

Example. *The maximum function* $\max(x,y)$ *on* $\{-1,1\}^2$ *has an expansion*

$$\max(x,y) = \frac{1}{2}(1 + x + y - xy)$$

which corresponds to the the conjunction function $f(x,y) = x \wedge y$ *on* $\{\top, \bot\}^2$*, or the minimum function on* $\mathbf{F}_2^2$*. To obtain this expansion, and a more general expansion for the maximum function* $\max_n$ *on n variables note that we can write*

$$\mathbf{I}(x = -1) = \frac{1}{2^n} \prod_{i=1}^n (1 - x_i) = \frac{1}{2^n} \sum_S (-1)^{|S|} x^S$$

[1] If a property on $\{-1,1\}^n$ is identified with it's indicator function valued on $\{0,1\}$, then the indicator function of $P \wedge Q$, for any two properties P and Q, is obtained by multiplying together the corresponding indicator functions. Since the property $x_i = y_i$ has indicator function $\mathbf{I}(x_i = y_i) = (1/2)(1 + x_i y_i)$, then we see that the expansion obtained for general functions $f : \{-1,1\}^n \to \{0,1\}$ is exactly the one induced by the canonical conjunctive form expression $P(x) = \bigvee_{P(y)} (y = x) = \bigvee_{P(y)} (\bigwedge_{i=1}^n y_i = x_i)$.

so that

$$\begin{aligned} max(x_1,\ldots,x_n) &= \mathbf{I}(x \neq -1) - \mathbf{I}(x = -1) \\ &= 1 - 2\mathbf{I}(x = -1) \\ &= \left(1 - \frac{1}{2^{n-1}}\right) - \frac{1}{2^{n-1}} \sum_{S \neq \emptyset} (-1)^{|S|} x^S \end{aligned}$$

Example. *Consider the minimum function* $\min(x) : \{-1,1\}^n \to \{-1,1\}$ *on n bits, which corresponds to the disjunction operation on* $\{\top,\bot\}^n$*. Since we have the relationship* $\min(x) = -\max(-x)$*, we find*

$$\begin{aligned} -\max(-x) &= -\left[\left(1 - \frac{1}{2^{n-1}}\right) - \frac{1}{2^{n-1}} \sum_{S \neq \emptyset} (-1)^{|S|} (-x)^S\right] \\ &= \left(\frac{1}{2^{n-1}} - 1\right) + \frac{1}{2^{n-1}} \sum_{S \neq \emptyset} x^S \end{aligned}$$

and this gives the expansion for $\min(x)$*.*

Note that the minimum and maximum functions on n variables are $1/2^{n-1}$ close to constant functions in L^1, hence the Fourier transformations of the minimum and maximum functions are $1/2^{n-1}$ close to the Fourier transforms of constant functions 1 and -1 in L^∞. Thus we see that the Fourier coefficients of min and max are bounded in absolute value by $1/2^{n-1}$, and the expected values of the functions are within a distance $1/2^{n-1}$ from 1 and -1.

Example. *In general, the indicator functions* $\mathbf{I}(x = a) : \{-1,1\}^n \to \{0,1\}$*, for some* $a \in \{-1,1\}^n$ *can be expressed as*

$$\mathbf{I}(x = a) = \frac{1}{2^n} \prod_{i=1}^{n} (1 + x_i a_i) = \frac{1}{2^n} \sum_S a^S x^S$$

If we only have a subset $X = \{i_1,\ldots,i_k\}$ *of indices we want to specify, then*

$$\mathbf{I}(x_{i_1} = a_1,\ldots,x_{i_k} = a_k) = \frac{1}{2^k} \sum_{S \subset X} a^S x^S$$

Thus the cardinality of sets S such that $\widehat{f}(S) \neq 0$ *is bounded by k.*

Example. *Consider n Bernoulli distributions over* $\{-1,1\}$ *with means* $\mu_1,\ldots,\mu_n$*, and consider the corresponding product distribution on* $\{-1,1\}^n$*, with density function* f*. If* X *is a random variable with this distribution, then*

$$\mathbf{P}(X_i = 1) - \mathbf{P}(X_i = -1) = 2\mathbf{P}(X_i = 1) - 1 = \mu_i$$

so $\mathbf{P}(X_i = 1) = \frac{1}{2}(\mu_i + 1) = P_i \in [0,1]$*. Then we can write*

$$\begin{aligned} f(x_1,\ldots,x_n) &= \prod_{i=1}^{n} [P_i \mathbf{I}(x_i = 1) + (1 - P_i)\mathbf{I}(x_i = -1)] \\ &= \sum_a P^{\{i:a_i=1\}}(1-P)^{\{i:a_i=-1\}}\mathbf{I}(x = a) \\ &= \frac{1}{2^n}\sum_S \sum_a a^S P^{\{i:a_i=1\}}(1-P)^{\{i:a_i=-1\}}x^S \\ &= \frac{1}{2^n}\sum_S (2P-1)^S x^S \\ &= \frac{1}{2^n}\sum_S \mu^S x^S \end{aligned}$$

where we obtain the second last equation by repeatedly factoring out the coordinates a_i *for* $a_i = 1$ *and* $a_i = -1$*.*

Example. *Consider the 'inner product mod 2' function* f *on* $\mathbf{F}_2^n \times \mathbf{F}_2^n$ *defined by*

$$f(x,y) = (-1)^{\langle x,y\rangle} = \prod_{i=1}^{n}(-1)^{x_i y_i}$$

Then $f(x,y)$ *measures the parity of the number of* x_i *which are equal to* y_i*. For* $n = 1$*, we have*

$$f(x,y) = (-1)^{xy} = \begin{cases} -1 & x = y = 1 \\ 1 & \textit{otherwise} \end{cases}$$

Hence, if we view f *as a function on* $\{-1,1\}^2$*, the induced function is just the* max *function on two variables, and has a Fourier representation*

$$f(x,y) = \frac{1}{2}(1 + x + y - xy)$$

In general, any set S corresponding to a set of indices on the functions of f can be associated with a unique pair of $S_1, S_2 \subset [n]$, corresponding to the indices relating to x on the left side of the function, and the indices relating to y on the right side of the equation. Let $\alpha(S)$ be the cardinality of $S_1 \cap S_2$. Then we have

$$\begin{aligned} f(x,y) &= \frac{1}{2^n}\prod_{i=1}^{n}(1+x_i+y_i-x_iy_i) \\ &= \frac{1}{2^n}\sum_{S\subset[n]}(-1)^{|S|}(xy)^S(1+x+y)^{S^c} \\ &= \frac{1}{2^n}\sum_{S\subset[n]}(-1)^{|S|}\sum_{T\subset S^c}(xy)^S(x+y)^T \\ &= \frac{1}{2^n}\sum_{S\in[n]^2}(-1)^{\alpha(S)}x^S \end{aligned}$$

Example. *The equality function $eq : \{-1,1\}^n \to \{0,1\}$ returns 1 only if all x_i are equal. Thus*

$$\begin{aligned} eq(x) &= \mathbf{I}(x_1 = x_2 = \cdots = x_n = 1) + \mathbf{I}(x_1 = x_2 = \cdots = x_n = -1) \\ &= \frac{1}{2^n}\sum_{S}(1+(-1)^{|S|})x^S = \frac{1}{2^{n-1}}\sum_{|S|\ even}x^S \end{aligned}$$

The not all equal function Neq returns 1 only if all x_i not all equal. Thus

$$Neq(x) = 1 - Eq(x) = \left(1 - \frac{1}{2^{n-1}}\right) - \sum_{\substack{|S|\ even \\ S\neq\varnothing}} x^S$$

1.2 Testing Linearity

A function $f : \{-1,-1\} \to \{-1,1\}$ is $\mathbf{F}_2$ linear if and only if $f(xy) = f(x)f(y)$ for all $x, y \in \{-1,1\}^n$, or equivalently, if there is $S \subset [n]$ such that $f(x) = x^S$. Given a general function f, there are effectively only two ways to explicitly check that the function is linear, we either check that $f(xy) = f(x)f(y)$ for all choices of the arguments x and y, or check that $f(x) = x^S$ holds for some choice of S, over all choices of x. Even assuming that f can be evaluated in

constant time, both methods take exponential time in n to compute[2]. The problem of testing linearity is a scenario in a family of problesm in the field of **property testing**, which attempts to design efficient algorithms to determine whether a particular boolean-valued function satisfies a certain property.

We might not be able to come up with a polynomial time algorithm to verify linearity, but we can make headway by considering the possiblity of coming up with an appropriate *randomized* algorithm which can verify linearity with high probability. The likely solution to the problem, given some function f, would to perform the linearity test for $f(XY) = f(X)f(Y)$ for a certain set of randomly chosen inputs X and Y. If f is non-linear, then $f(XY) \neq f(X)f(Y)$ with positive probability, and if we find this particular input we can guarantee the function is non-linear. If f is linear, the test always passes. We shall find that the probability that the test fails directly relates to how similar a function is to a linear function.

To proceed along these lines, we must introduce the probabilistic interpretation of the Fourier expansion. The Fourier coefficients of $f : \mathbf{F}_2^n \to \mathbf{R}$ can, of course, be calculated as the inner product

$$\widehat{f}(S) = \langle f, x^S \rangle = \frac{1}{2^n} \sum_x f(x) x^S$$

where the 2^n factor is included so that characters are normalized. We can also see this as a probabilistic statement, if we consider the uniform distribution on $\{-1, 1\}^n$, and consider the corresponding random variable X, then we can write

$$\widehat{f}(S) = \mathbf{E}[f(X)X^S]$$

So that the Fourier coeffient measures the average value of f, relative to the parity over S. Note that this implies that a boolean-function $f : \{-1, 1\}^n \to \{-1, 1\}^n$ is *unbiased*, that is, it has an equal chance of taking -1 and 1, if and only if $\widehat{f}(\varnothing) = 0$. Of course, we have Parseval's equality

$$\mathbf{E}[f^2(X)] = \sum_S \widehat{f}(S)^2$$

[2]This is effectively guaranteed, because the description length of f is always exponential in n, and if an algorithm determining linearity does not check the entire description of a linear function f, we can modify f to a non-linear function on inputs that the linearity tester doesn't look at, at the algorithm will not be able to distinguish between these two inputs.

and so

$$\mathbf{V}[f(X)] = \mathbf{E}[f^2(X)] - \mathbf{E}[f(X)]^2 = \sum_{S \neq \emptyset} \widehat{f}(S)^2$$

and similarily,

$$\mathrm{Cov}[f(X), g(X)] = \sum_{S \neq \emptyset} \widehat{f}(S)\widehat{g}(S)$$

So that the Fourier coefficients efficiently measure probabilistic quantities.

If f and g are *boolean-valued* maps $\mathbf{F}_2^n \to \{-1, 1\}$, then

$$\begin{aligned}\mathbf{E}[f(X)g(X)] &= \mathbf{P}(f(X) = g(X)) - \mathbf{P}(f(X) \neq g(X)) \\ &= 1 - 2\mathbf{P}(f(X) \neq g(X))\end{aligned}$$

Note that $\mathbf{P}(f(X) \neq g(X))$ differs from the Hamming distance of f and g by a constant factor. We define $\mathbf{P}(f(X) \neq g(X))$ to be the relative hamming distance between f and g, denoted $d(f, g)$. Since $f^2 = 1$, this implies $\|f\|_2 = 1$, and therefore

$$\begin{aligned}\mathbf{V}(f) &= \mathbf{E}[f(X)^2] - \mathbf{E}[f(X)]^2 \\ &= 1 - \mathbf{E}[f(X)]^2 \\ &= 1 - (\mathbf{P}(f(X) = 1) - \mathbf{P}(f(X) = -1))^2 \\ &= 4\mathbf{P}(f(X) = -1)\mathbf{P}(f(X) = 1) \in [0, 1]\end{aligned}$$

so the variance of a boolean-valued function is very closely related to the degree to which the function is constant.

Lemma 1.1. *If* $\varepsilon = \min[d(f, 1), d(f, -1)]$, *then* $2\varepsilon \leqslant \mathbf{V}(f) \leqslant 4\varepsilon$.

Proof. We are effectively proving that for any $x \in [0, 1]$,

$$2\min(x, 1 - x) \leqslant 4x(1 - x) \leqslant 4\min(x, 1 - x)$$

Since $4x(1-x) = 4\max(x, 1-x)\min(x, 1-x)$, we may divide by $4\min(x, 1 - x)$ to restate the inequality as

$$1/2 \leqslant \max(x, 1 - x) \leqslant 1$$

which is obvious. □

This shows that if we have a sequence of functions $f_i : \{-1,1\}^n \to \{-1,1\}$, then $\varepsilon_i \sim x_i$ holds if and only if $\mathbf{V}(f) \sim x_i$. The minimum hamming distance to a constant value is asymptotically the same as the variation of the function. In general domains we intuitively view $\mathbf{V}(f)$ as a measure of how constant the function f is, but for functions $\{-1,1\}$ we have a precise, quantitative estimate.

Example. *Consider a function f chosen uniformly randomly from the set of all Boolean functions on $\{-1,1\}^n$. That is, for any function $g : \{-1,1\}^n \to \{-1,1\}$, the probability distribution gives $\mathbf{P}(f = g) = 1/2^{(2^n)}$. Since each Boolean function has a Fourier expansion, f has a Boolean expansion*

$$f(x) = \sum \widehat{f}(S)x^S$$

where each $\widehat{f}(S) \in \mathbf{R}$ is now a random variable. Now because we have a bijection amongst the set of all Boolean functions, mapping g to $-g$, we find

$$\begin{aligned}\mathbf{E}[\widehat{f}(S)] &= \sum \mathbf{P}(f = g)\widehat{g}(S) = \sum \mathbf{P}(f = g)\widehat{(-g)}(S) \\ &= -\sum \mathbf{P}(f = g)\widehat{g}(S) = -\mathbf{E}[\widehat{f}(S)]\end{aligned}$$

Hence $\mathbf{E}[\widehat{f}(S)] = 0$. What's more,

$$\begin{aligned}\mathbf{V}[\widehat{f}(S)] = \mathbf{E}[\widehat{f}(S)^2] &= \mathbf{E}\left[\mathbf{E}\left(f(X)X^S\right)^2\right] \\ &= \mathbf{E}[f(X)f(Y)(XY)^S] \\ &= \mathbf{P}(X = Y) + \mathbf{P}(X \neq Y)\mathbf{E}[f(X)f(Y)(XY)^S | X \neq Y] \\ &= \frac{1}{2^n} + \left(1 - \frac{n}{4^n}\right)(2\mathbf{P}(f(X) = f(Y)|X \neq Y) - 1)\,\mathbf{E}[(XY)^S | X \neq Y]\end{aligned}$$

For any particular $x \neq y$, $\mathbf{P}[f(x) = f(y)] = 1/2$, because f is chosen uniformly from all functions, and for any function g, we have the function g', such that $g'(z) = g(z)$ except when $z = x$, where $g'(z) = -g(x)$, and the association is a bijection. This implies that $\mathbf{P}(f(X) = f(Y)|X \neq Y) = 1/2$, hence the variation satisfies $\mathbf{V}[\widehat{f}(S)] = 1/2^n$. Thus functions on average have no Fourier coefficient on each S, and most functions have very small Fourier coefficient.

Since a function $f : \{-1,1\}^n \to \{-1,1\}$ always has square-norm 1, because $f^2 = 1$, Parsevel's theorem implies that $\sum \widehat{f}(S)^2 = 1$ if we sum over

all the Fourier coeffients, so that a boolean-valued function on n variables gives rise to a probability distribution over $[n]$. We call this the spectral sample of f, and denote the distribution by $\mathcal{S}_f$.

Often times, the Fourier coeffients of sets over some particular cardinality are more than enough to determine some result. We define the weight of a function $f : \{-1,1\}^n \to \mathbf{R}$ of degree k to be

$$\mathbf{W}^k(f) = \sum_{|S|=k} \widehat{f}^2(S)$$

If f is boolean-valued, then we can also define $\mathbf{W}^k(f) = \mathbf{P}(|S| = k)$ where S is a set randomly drawn from the spectral sample of f. $\mathbf{W}^k(f)$ is also the square norm of the function

$$(P^k f)(x) = \sum_{|S|=k} \widehat{f}(S) x^S$$

which can be seen as a certain truncation estimate of f.

The simplest version of linearity testing runs using one random query as a test – it just takes one pair of inputs (X, Y), and tests the linearity of this function against these inputs. This is known as the Blum-Luby-Rosenfeld algorithm, or BLR for short. It turns out that the sucess of this method is directly related to how linear a function is. Define a function to be **approximately linear** if

- $f(x+y) = f(x) + f(y)$ for a large majority of inputs x and y.
- There is a linear functional $g : \mathbf{F}_2^n \to \mathbf{F}_2$ such that $f(x) = g(x)$ for a large number of inputs x.

It is clear that the second property implies the first, but not that the first implies the second in a precise relationship. To be more precise, we say that f is ε-close to being linear if there is a linear g with $d(f,g) \leqslant \varepsilon$. What the analysis of BLR shows is that $f(X+Y) = f(X) + f(Y)$ holds with probability ε, then f is ε close to a linear function. Since we are applying Hamming distances in measuring how linear a function is, we can guess that the Harmonic analysis of boolean functions will come in handy.

Theorem 1.2. *Given a function $f : \mathbf{F}_2^n \to \mathbf{F}_2$, if the BLR algorithm accepts f with probability greater than $1 - \varepsilon$, then f is at least ε-close to being linear.*

Proof. We switch to multiplicative notation, so that $f : \{-1,1\}^n \to \{-1,1\}$. Note that the probability that $f(XY) = f(X)f(Y)$ is the same as the probability that

$$\frac{1}{2}[1 + f(X)f(Y)f(XY)] = 1$$

and if $f(XY) \neq f(X)f(Y)$, then

$$\frac{1}{2}[1 + f(X)f(Y)f(XY)] = 0$$

so that the function $g(x,y) = (1/2)[1 + f(x)f(y)f(xy)]$ is 0-1 valued, and

$$\begin{aligned} 1 - \varepsilon &= \mathbf{P}[\text{BLR accepts } f] \\ &= \mathbf{E}[g(X,Y)] \\ &= \frac{1}{2} + \frac{1}{2}\mathbf{E}_X[f(X)\mathbf{E}_Y[f(Y)f(XY)]] \\ &= \frac{1}{2} + \frac{1}{2}\mathbf{E}_X[f(X)(f * f)(X)] \\ &= \frac{1}{2} + \frac{1}{2}\sum \widehat{f}(S)^3 \end{aligned}$$

and therefore

$$1 - 2\varepsilon = \sum \widehat{f}(S)^3 \leqslant \|\widehat{f}\|_\infty \|f\|_2 = \|\widehat{f}\|_\infty$$

Now $\widehat{f}(S) = 1 - 2d(f, x^S)$, and since there is S' with $\widehat{f}(S') \geqslant 1 - 2\varepsilon$, this implies $d(f, x^{S'}) \leqslant \varepsilon$, and therefore f is ε-close to the linear function $x^{S'}$. □

Note that this algorithm, with only three evaluations of the function f, can determine with high accuracy that f is not linear, if the function is far away from being non-linear to begin with, or, if we are wrong, then f is very similar to a linear function in the first case. Yet given f, we cannot use this algorithm to determine the linear function x^S which f is similar to. Of course, for most x, $f(x) = x^S$. However, we cannot use this estimate to obtain accurate estimates in expectation for a fixed x, in the sense that there is no measure of the likelihood of the estimate being equal to the actual value. The problem is that x is a fixed quantity, rather than varying over many values of the function, and is therefore our test is easy to break. Note, however, that $x^S(x) = x^S(xy)x^S(y)$, and if we let y be a random quantity, then $x^S(xy)$ and $x^S(y)$ will likely be equal to $f(x+y)$ and $f(y)$ with a probability that is feasible to determine.

Theorem 1.3. *If $f : \mathbf{F}_2^n \to \{-1, 1\}$ is ε-close to x^S, then for any y,*

$$\mathbf{P}_X[f(X+y)f(X) = x^S(y)] \geqslant 1 - 2\varepsilon$$

Proof. The probability that $f(X+y) \neq x^S(X+y)$ is less than ε, as is the probability that $f(X) \neq x^S(X)$. By the union bound, this implies that the probability of either occuring is less than or equal to 2ε. But then the probability that both do not occur is greater than or equal to $1 - 2\varepsilon$, and when this occurs, we can guarantee that $f(X+y)f(X) = x^S(y)$, thus our estimate $f(X+y)f(X)$ is equal to $x^S(y)$ with probability $\geqslant 1 - 2\varepsilon$. □

Thus linear functions are *locally correctable*, in that we can correct small modifications to linear functions with high probability. This turns out to have vast repurcussions in complexity theory, which we will discuss later. One reason why these methods work is that the linear functions on $\mathbf{F}_2^n$ form a very discrete set. For $S \neq T$,

$$d(x^S, x^T) = \mathbf{P}(x^S \neq x^T) = \mathbf{P}(x^{S\Delta T} = -1) = 1/2$$

Thus characters are separated by a dimension-independant quantity.

To recapitulate why Fourier analysis applies to this problem in another light, the translation operators

$$(\sigma_y f)(x) = f(yx)$$

are diagonalized by the characters x^S, which form an orthogonal basis such that

$$\sigma_y x^S = (yx)^S = y^S x^S$$

Thus the equation that $f(xy) = f(x)f(y)$ holds for all x, y can be rephrased by the diagonalization argument that $\sigma_y f = \sigma_{f(y)} f$, and this holds only if $y^S \widehat{f}(S) = f(y)\widehat{f}(S)$, where $f(x) = \sum \widehat{f}(S)x^S$, so that $f(y) = y^S$ or $\widehat{f}(S) = 0$. We have $2d(f,g) = \|f - g\|_1$, so that $\|\widehat{f} - \widehat{g}\|_\infty = 2d(f,g)$. In a similar light, if a Boolean-valued function f differs from a linear function x^S with probability ε, the standard L^1-L^∞ equivalence shows that

$$|\widehat{f}(T)| \leqslant 2\varepsilon$$

for $T \neq S$, and $|\widehat{f}(S) - 1| \leqslant 2\varepsilon$, and the converse also holds provided we assume ahead of time that f is Boolean valued. Using this, we can obtain

a tighter bound on the converse of the statement, that for small enough ε, if a function f is ε-close to some linear function x^S, then the BLR test accepts f with probability greater than $1 - 3\varepsilon + O(\varepsilon^2)$, and this bound is tight up to first order.

Theorem 1.4. *If f is ε-close to some linear function x^S, then the BLR test rejects f with probability at least $3\varepsilon - 10\varepsilon^2 + 8\varepsilon^3$.*

Proof. Note that

$$\widehat{f}(S) = \mathbf{E}[f(X)X^S] = 1 - 2d(f, x^S) = 1 - 2\varepsilon$$

Hence

$$\begin{aligned}
\mathbf{P}(\text{BLR rejects } f) &= \mathbf{E}\left(\frac{1 - f(X)f(Y)f(XY)}{2}\right) \\
&= \frac{1}{2} - \frac{1}{2}\sum_{T \neq S} \widehat{f}(T)^3 - \frac{1}{2}\widehat{f}(S)^3 \\
&\geqslant \frac{1}{2} - \varepsilon \sum_{T \neq S} \widehat{f}(T)^2 - \frac{1}{2}\widehat{f}(S)^3 \\
&= \frac{1}{2} - \varepsilon(1 - \widehat{f}(S)^2) - \frac{1}{2}\widehat{f}(S)^3 \\
&= 3\varepsilon - 10\varepsilon^2 + 8\varepsilon^3
\end{aligned}$$

This is the best upper bound we can obtain, up to a first order. For each n, consider the function $g_n : \mathbf{F}_2^n \to \mathbf{F}_2$, with

$$g_n(x) = \mathbf{I}(x = 1)$$

Let us count all instances when g_n is rejected by the BLR test. There are three possibilities where $g_n(x + y) \neq g_n(x) + g_n(y)$:

- If $x = 1$ and $y \neq 1$.
- If $x \neq 1$ and $y = 1$.
- If $x \neq 1$ and $y \neq 1$, and $x + y = 1$.

There are $2^n - 1$ instances of the first and second rejection, and $2^n - 2$ instances of the third. Thus

$$\mathbf{P}(\text{BLR rejects } g_n) = \frac{2(2^n - 1) + (2^n - 2)}{4^n} = \frac{3}{2^n} - \frac{4}{4^n}$$

and g_n is $1/2^n$ close to 0. If there was a lower bound of the form $C\varepsilon - O(\varepsilon^2)$, for $C > 3$, and if we write $1/2^n = \varepsilon$, then for big enough n we find

$$3\varepsilon - 4\varepsilon^2 \geqslant (3 + \delta)\varepsilon$$

Hence

$$-4\varepsilon \geqslant \delta$$

which is impossible since we can let ε tend to zero. □

The converse of this statement is that if the BLR test accepts f with probability greater than $1 - \varepsilon + O(\varepsilon^2)$, then f is $\varepsilon/3$ close to being linear. Thus for small ε the bound on similarity to a linear function is even more tight than we initially calculated.

1.3 The Walsh-Fourier Transform

Historically, the study of the Fourier transform on $\mathbf{F}_2^n$ emerged from the Fourier theory of $L^2[0,1]$. Consider the product group $\mathbf{F}_2^\infty = \prod \mathbf{F}_2$, which is a compact abelian group. We have the standard projects $i_n : \mathbf{F}_2^n \to \mathbf{F}_2^\infty$, defined by $i_n(x_1, \dots, x_n) = (x_1, \dots, x_n, 0, 0, \dots)$. In addition, we have the projections $\pi_n(x_1, x_2, \dots) = (x_1, \dots, x_n)$. Given a character f on $\mathbf{F}_2^\infty$, each $f \circ i_n$ is a character on $\mathbf{F}_2^n$, and can thus be written χ_{S_n} for some $S_n \subset [n]$. The sets S_n defined must be increasing in n, and for $n \leqslant m$, satisfy $S_m \cap [n] = S_n$. For any $x = (x_1, x_2, \dots)$ in $\mathbf{F}_2^\infty$, the sequence $\pi_n(x)$ converges to x, and hence

$$f(x) = \lim \pi_n(x) = \lim x^{S_n}$$

If $\bigcup S_n$ is not a bounded subset of the integers, then the equation above says f cannot be a continuous character on $\mathbf{F}_2^\infty$. Thus the characters on $\mathbf{F}_2^\infty$ correspond to finite subsets of positive integers, and are essentially the inductive limit of the character groups $(\mathbf{F}_2^n)^*$.

An important link between probability theory and real analysis is that the binary expansion of real numbers on $[0,1]$,

$$x = \sum_{m=1}^{\infty} \frac{x_m}{2^m}$$

can be seen as defining a measure preserving map between $\mathbf{F}_2^\infty$ and $[0,1]$, where

$$(x_1, x_2, \dots) \mapsto \sum_{m=1}^{\infty} \frac{x_m}{2^m}$$

which is injective almost everywhere. Thus we can find a uniform distribution over $[0,1]$ by taking the binary expansion obtained by flipping a fair coin infinitely many times.

Returning to our interests, the Harmonic analysis of $\mathbf{F}_2^\infty$ gives rise to an expansion theory on $[0,1]$. In particular, we find that the characters χ_S, which form an orthonormal basis for $L^2(\mathbf{F}_2^\infty)$ correspond to an orthonormal basis of functions on $L^2[0,1]$ of the form

$$w^S(x) = \chi_S(x_1, x_2, \dots) = \prod_{i \in S} (-1)^{x_i} = \prod_{i \in S} r_i(x)$$

where $r_i(x) = (-1)^{x_i}$ is the i'th Rademacher function. Because integers have unique binary expansions, we may reindex w^S as w_n for a unique non-negative integer n, as was done classically, and this family of functions are called the Walsh functions. Thus every function has a unique expansion

$$f(x) = \sum_{n=0}^{\infty} a_n w_n(x) = \sum_{\substack{S \subset [n] \\ \#S < \infty}} \widehat{f}(S) w^S(x)$$

and this is essentially where the beginnings of the study of Boolean expansions began, by Walsh in the 1920s. These expansions have very good L^p truncations, but they are less used in modern mathematics because the functions w_n are not smooth, and as such are not useful in the study of partial differential equations.

For our purposes, we shall be interested in this system in order to come up with a computationally efficient way of computing the Walsh-Fourier

expansion of a boolean valued function $f : \mathbf{F}_2^n \to \mathbf{R}$. For an integer k, define k_n to be the coeffient in the binary expansion

$$k = \sum k_n 2^n$$

First, note that there is a basis $\{X_i\}$ of functions from $\mathbf{F}_2^n \to \mathbf{R}$ defined by

$$X_k = \mathbf{I}(x_1 = k_1, \ldots, x_n = k_n)$$

and a basis $\{Y_i\}$ of elements of $\mathbf{R}\langle \mathbf{F}_2^* \rangle$ defined by

$$Y_k = \chi_{\{i:k_i=1\}}$$

Since the Walsh transform is effectively a map from $\mathbf{R}^{\mathbf{F}_2}$ to $\mathbf{R}\langle \mathbf{F}_2^* \rangle$, these bases induce a matrix representation of the transform. We have

$$\mathcal{W}(X_n) = \frac{1}{2^n} \sum_S (-1)^{\sum_{m \in S} k_m} \chi_S$$

Hence if we define the matrices

$$H_1 = \begin{pmatrix} 1 \end{pmatrix}$$

$$H_{n+1} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}$$

Then the matrix representation of $\mathcal{W}$ is $H_n/2^n$, because, as can be proved by induction, the (a,b)'th entry of H_n is $(-1)^{\sum a_i b_i}$. By a divide and conquer approach, if we write a vector in $\mathbf{R}^{\mathbf{F}_2^n}$ as (v,w), where v and w split the space in half according to the basis defined above, then

$$H_{n+1}(v,w) = (H_n v + H_n w, H_n v - H_n w)$$

Using this approach, if t_n is the number of additions and subtractions required to calculate the Walsh transform a n-dimensional Boolean function, then $t_{n+1} = 2t_n + 2n$, with $t_1 = 0$, then

$$t_n = 2^{n+1} \sum_{k=2}^{n} k \left(\frac{1}{2}\right)^k \leqslant n 2^{n+1} \sum_{k=2}^{n} \left(\frac{1}{2}\right)^k \leqslant n 2^n$$

so the calculation is possible in $n2^n$ calculations, which looks bad, but the algorithm is polynomial in the size of the input, because an element of $\mathbf{R}^{\mathbf{F}_2^n}$ takes $m = 2^n$ numbers to specify, so the algorithm is $m \lg m$. This is essentially the fastest way to compute the Fourier coefficients of an arbitrary boolean function.

Chapter 2

Social Choice Functions

In this chapter, we interpret boolean functions in the context of the theory of social choice. Here we think of a boolean function $f : \{-1,1\}^n \to \{-1,1\}$ as a voting rule; n people choose between two candidates, labelled -1 and 1, and the function f determines the overall outcome of the vote. In this scenario is is natural to try and find fair voting rules, and this leads to classifications of Boolean functions by properties relating to voting principles, and we will find these properties naturally have interest in the harmonic analysis of boolean functions in other general contexts.

The standard voting rule over an odd number of candidates is the Majority function $\mathrm{Maj}_n : \{-1,1\}^n \to \{-1,1\}$, for n odd, which choose the candidate with the majority of votes. A function over even n is called a majority function if it always chooses the candidate with the majority of votes when possible. But there are certainly other voting rules which are applied in practice.

Example. *The voting rule $And_n : \{-1,1\}^n \to \{-1,1\}$ corresponds to choosing candidate 1 unless every other person votes for candidate -1. The rule $Or_n : \{-1,1\}^n \to \{-1,1\}$ votes for candidate -1 unless everyone else votes against him.*

Example. *The dictator rule $\chi_i : \{-1,1\}^n \to \{-1,1\}$ defined by $\chi_i(x) = x_i$ places the power of decision making in a single person.*

Example. *All of these functions can be quantified as a version of the weight majority, or linear threshold function, those functions f which can be defined, for some $a_0, \ldots, a_n \in \mathbf{R}$, as*

$$f(x) = sgn(a_0 + a_1 x_1 + \cdots + a_n x_n)$$

Thus each voter has a certain voting power, specified by the a_i, and there is an additional bias a_0.

Example. *In the United States, a recursive majority voting rule is used. A simple version of the rule is defined on n^d bits by*

$$Maj_n^{\otimes d}(x_1,\dots,x_d) = Maj_n(Maj_n^{\otimes(d-1)}(x_1), Maj_n^{\otimes(d-1)}(x_n))$$

where each $x_i \in \{-1,1\}^{n^{d-1}}$. Thus we subdivide voters into certain regions, determine the majority over this region, and then take the majority over the accumulated choices.

Example. *The tribes function divides voters into tribes, and the outcome of the vote holds if and only if one tribe is unanimously in favour of the vote. The width w, size s tribe voting rule is defined by*

$$Tribes_{ws} : \{-1,1\}^{sw} \to \{-1,1\}$$

$$Tribes_{ws}(x_1,\dots,x_s) = Or_s(And_w(x_1),\dots,And_w(x_s))$$

where each $x_i \in \{-1,1\}^w$.

Any boolean function is a voting rule on two candidates, so the study of voting functions is really just a language for talking about general properties of boolean functions. However, it certainly motivates the development of certain contexts which will become very useful in the future. For instance, important properties of Boolean functions become desirable properties of voting rules. In particular,

- A voting rule f is **monotone** if $x \leqslant y$ implies $f(x) \leqslant f(y)$.
- A voting rule is **odd** if $f(-x) = -f(x)$.
- A rule is **unanimous** if $f(1) = 1$, and $f(-1) = -1$.
- A voting rule is **symmetric** if $f(x^\pi) = f(x)$, where $\pi \in S_n$ acts on $\{-1,1\}^n$ by permuting coordinates.
- A voting rule is **transitive symmetric** if, for any index i, there is a permutation π taking i to any other index j, such that $f(x^\pi) = f(x)$.

There is only a single function which satisfies all of these properties for odd n, the majority function Maj_n. This constitutes May's theorem.

Theorem 2.1. *The majority function is the only monotone, odd, unanimous, symmetric, transitive symmetric function in odd dimension. In even dimensions, such a function does not exist.*

Proof. Suppose $f : \{-1,1\}^n \to \{-1,1\}$ is a function which is monotone and symmetric. Then $f(x) = g(\#\{i : x^i = 1\})$ for some monotone function $g : \{0,\ldots,n\} \to \{-1,1\}$. The monoticity implies that there is N for which

$$g(n) = \begin{cases} -1 & n < N \\ 1 & n \geqslant N \end{cases}$$

Note that if f is an odd function, then

$$\mathbf{E}[f(X)] = \mathbf{E}[f(-X)] = -\mathbf{E}[f(X)]$$

hence $\mathbf{E}[f(X)] = 0$, so f takes $+1$ and -1 with equal probability. But also

$$0 = \mathbf{E}[f(X)] = \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} g(k)$$

Hence

$$\sum_{k=0}^{N-1} \binom{n}{k} = \sum_{k=N}^{n} \binom{n}{k}$$

Since the left side strictly increases as a function of N, and the right side strictly decreases, if a N exists which satisfies this equality it is unique. If n is odd, then we can pick $N = (n+1)/2$, because the identity

$$\binom{n}{k} = \binom{n}{n-k}$$

which implies coefficients can be paired up. If n is even, the same pairing technique shows that such a choose of N is impossible. □

2.1 Influence

Given a voting rule $f : \{-1,1\}^n \to \{-1,1\}$, we may wish to quantify the power of each voter in changing the outcome of the vote. If we choose a particular voter i and we fix all inputs but the choice of the i'th voter,

we say that the voter i is **pivotal** if his choice affects the outcome of the election. That is, givne $x \in \{-1,1\}^n$, we define $x^{\oplus i} \in \{-1,1\}^n$ to be the input obtained by flipping the i'th bit. Then i is pivotal on x exactly when $f(x^{\oplus i}) \neq f(x^i)$. It is important to note that even if a voting rule is symmetric, it does not imply that each person has equal power over the results of the election, once other votes have been fixed (For instance, in a majority voitng system with 3 votes for candidate A and 4 votes for candidate B, the voters for B have more power because their choice, considered apart from the others seriously impacts the result of the particular election).

It is natural to define the *overall power* of a voter i on a certain voting rule f as the likelihood that the voter is pivotal on the election. We call this the **influence** of the voter, denoted $\mathrm{Inf}_i(f)$. To be completely accurate, we would have to determine the probability that each particular choice of votes occurs (If there is a high chance that one candidate is going to one, we should expect each voter to individually have little power, whereas if the competition is tight, each voter should have much more power). We assume that all voters choose outcomes independently, and uniformly randomly (this is the **impartial culture** assumption), so that that the influence takes the form

$$\mathrm{Inf}_i(f) = \mathbf{P}(f(X) \neq f(X^{\oplus i}))$$

Thus the influence measures the probability that index i is pivotal uniformly across all elements of the domain.

The impartial culture assumption is not only for simplicity, in the face of no other objective choice of distribution, but also because the formula then has combinatorial interest. If we colour a node on the Hamming cube based on the output of the function f, then the influence of an index i is the fraction of coordinates whose color changes when we move along the i'th dimension of the cube. Similarily, we can also consider it as the fraction of edges along the i'th dimension upon which f changes.

Example. *The dictator function χ_i satisfies $Inf_i(\chi_i) = 1$, and $Inf_j(\chi_i) = 0$ otherwise. If f is an arbitrary voting rule for which $Inf_i(f) = 0$, then f completely ignores index i, we can actually specify f as a function of the remaining $n-1$ variables.*

Example. *The i'th index is pivotal for the And_n function only at the points 1 and $1^{\oplus i}$, so that*

$$Inf_i(And_n) = 2/2^n = 2^{1-n}$$

Example. *To calculate the influence of Maj_n, we note that if $Maj_n(x) = -1$, and $Maj_n(x^{\oplus i}) = 1$, then there are $(n+1)/2$ indices j such that $x^j = -1$, and i is one of these indices. Once i is fixed, the total possible choices of indices in which these circumstances hold is*

$$\binom{n-1}{\frac{n-1}{2}}$$

hence if we also consider x for which $Maj_n(x) = 1$, and $Maj_n(x^{\oplus i}) = -1$, then

$$Inf_i(Maj_n) = 2\binom{n-1}{\frac{n-1}{2}}/2^n = \binom{n-1}{\frac{n-1}{2}}2^{1-n}$$

Applying Stirling's approximation, we can compute the approximation

$$Inf_i(Maj_n) = \sqrt{\frac{2}{n\pi}} + O(n^{-3/2})$$

We will soon show this is about the best influence we can find for a monotonic, symmetric voting rule.

To connect the influence of a boolean function to it's Fourier expansion, we must come up with an analytic expression for the influence. Consider the i'th partial derivative operator

$$(D_i f)(x) = \frac{f(x^{i\to 1}) - f(x^{i\to -1})}{2}$$

In this equation, we see a slight relation to differentation of real-valued functions, and the analogy is completed when we see how D_i acts on the polynomial representation of f. Indeed,

$$D_i x^S = \begin{cases} x^{S-\{i\}} & : i \in S \\ 0 & : i \notin S \end{cases}$$

Hence by linearity,

$$(D_i f)(x) = \sum_{i \in S} \widehat{f}(S) x^{S-\{i\}}$$

For Boolean-valued functions f, $D_i f$ is intimately connected to the influence of f, because

$$(D_i f)(x) = \begin{cases} \pm 1 & f(x^i) \neq f(x^{\oplus i}) \\ 0 & f(x^i) = f(x^{\oplus i}) \end{cases}$$

Hence $(D_i f)^2(x)$ is the 0-1 indicator of whether i is pivotal at x, and so

$$\mathrm{Inf}_i(f) = \mathbf{E}[(D_i f)^2] = \|D_i f\|_2^2 = \sum_{i \in S} \widehat{f}(S)^2$$

Defining the i'th **Laplacian** operator $L_i = x_i D_i$, we find

$$(L_i f)(x) = \sum_{i \in S} \widehat{f}(S) x^S$$

Hence L_i is a projection operator, satisfying $\mathrm{Inf}_i(f) = \langle L_i f, L_i f \rangle = \langle f, L_i f \rangle$. Thus we have found a quadratic representation of $\mathrm{Inf}_i(f)$.

We shall dwell on the Laplacian for slightly longer than the other operators, since we will find it's generalization occuring in future discussion. As a projection operator, it has a complementary projection

$$(E_i f)(x) = \frac{f(x^{i \to 1}) + f(x^{i \to -1})}{2}$$

which can be considered the expected value of f, if we fix all coordinates except for the i'th index, which takes a uniformly random value. Thus for any boolean function f we have

$$f = E_i f + x_i D_i f$$

Both $E_i f$ and $D_i f$ do not depend on the i'th coordinate of f, which is useful for carrying out inductions on the dimension of a boolean function's domain. We also note that we have a representation of the Laplacian as a difference

$$(L_i f)(x) = \frac{f(x) - f(x^{\oplus i})}{2}$$

so that the Laplacian acts very similarily to D_i.

Though we use the fact that $(D_i f)^2 = |D_i f|$ to obtain a quadratic equation for $D_i f$, the definition of the influence as the absolute value $|D_i f|$ as the definition of influence still has its uses, especially when obtaining L^1 inequalities for the influence. Indeed, we can establish the bound

$$\mathrm{Inf}_i(f) = \mathbf{E}|D_i f| \geqslant |\mathbf{E}(D_i f)| = |\widehat{f}(i)|$$

This inequality only becomes an equality when f is always increasing in the i'th direction, or always decreasing – that is, if $D_i f \geqslant 0$ or $D_i f \leqslant 0$,

and if $D_i f \geqslant 0$ we in fact have $\mathrm{Inf}_i(f) = \widehat{f}(i)$, since $\widehat{f}(i)$ is non-negative. In particular, if f is monotone, then we have $\mathrm{Inf}_i(f) = \widehat{f}(i)$ for all i, which is very useful. As a first application of this fact, we show that in a monotone, transitive symmetric voting system, all voters have small influence.

Theorem 2.2. *Given a monotone, transitive symmetric $f : \{-1,1\}^n \to \{-1,1\}$,*

$$Inf_i(f) \leqslant \frac{1}{\sqrt{n}}$$

Proof. Applying Parsevel's inequality, we find

$$1 = \|f\|_2^2 = \sum \widehat{f}(S)^2 \geqslant \sum \widehat{f}(i)^2 = n\widehat{f}(i)^2$$

Hence $\widehat{f}(i) \leqslant 1/\sqrt{n}$. But by monotonicity,

$$\mathrm{Inf}_i(f) = \widehat{f}(i)$$

Putting these equations together gives us the inequality. □

The **total influence** of a boolean function $f : \{-1,1\}^n \to \{-1,1\}$, denoted $\mathbf{I}(f)$, is the sum of the influences $\mathrm{Inf}_i(f)$ over each coordinate. It is not a measure of how powerful each voter is, but instead how chaotic the system is with respect to how any of the voters change their coordinates. For instance, constant functions have total influence zero, where no vote change effects the outcome in any way, whereas the function $x^{[n]}$ has total influence n, since every change in a voters choices flips the outcome of the vote. For monotone transitive symmetric voting rules, we already know the total influence is bounded by $\sqrt{n}$, and this is asymptotically obtained by the majority function.

Example. *The total influence of χ_S is S, because all the indices of S are pivotal for all inputs. The total influence is maximized over all functions by the parity function $x^{[n]}$, which is always pivotal for all inputs. The total influence of And_n and Or_n is $n2^{1-n}$, rather small, whereas Maj_n has total influence*

$$\sqrt{2n/\pi} + O\left(\sqrt{1/n}\right)$$

which is the maximum total influence of any monotone transitive symmetric function.

As a sum of quadratic forms, the total influence is also a quadratic form, but cannot be represented by a projection. Considering

$$\mathbf{I}(f) = \sum_i \mathrm{Inf}_i(f) = \sum_i \langle f, L_i f \rangle = \left\langle f, \left(\sum L_i \right) f \right\rangle$$

we see that the total influence is represented as a quadratic form over the **Laplacian** operator $L = \sum L_i$. Note that

$$(Lf)(x) = \sum_{i=1}^{n} \sum_{i \in S} \widehat{f}(S) x^S = \sum |S| \widehat{f}(S) x^S$$

so the Laplacian amplifies the coefficients of f corresponding to sets of large weight. This makes sense, because x^S changes on more inputs when S contains more indices, hence the function should have higher influence. This implies that

$$\mathbf{I}(f) = \sum |S| \widehat{f}(S)^2 = \sum k \mathbf{W}^k(f)$$

for Boolean-valued functions, the total influence is the expected cardinality of $\mathcal{S}$, where $\mathcal{S}$ is a random set distributed according to the spectral distribution induced from f.

There is a probabilistic interpretation of the total influence, which makes it interesting to the theory of voting systems. Define $\mathrm{sens}_f(x)$ to be the number of pivotal indices for f at x. Then $\mathbf{I}(f) = \mathbf{E}[\mathrm{sens}_f(X)]$. Indeed, we find

$$(Lf)(x) = \sum_i \frac{f(x) - f(x^{\oplus i})}{2} = \frac{n}{2}[f(x) - \mathbf{E}[f(x^{\oplus i})]]$$

where the expectation is taken where i is random, uniformly distributed over all indices $[n]$. We have

$$\mathbf{E}[f(x^{\oplus i})] = \frac{n - \mathrm{sens}_f(x)}{n} f(x) + \frac{\mathrm{sens}_f(x)}{n}(-f(x)) = f(x) \frac{n - 2\mathrm{sens}_f(x)}{n}$$

Hence $(Lf)(x) = f(x)\mathrm{sens}_f(x)$.

There is also a combinatorial interpretation of the total influence. Each vertex has n edges, and if we consider the probability distribution which

first picks a point on the cube uniformly, and then an edge uniformly, then the resulting distribution will be the uniform distribution on edges. Since the expected number of **boundary edges** (those (x, y) for which $f(x) \neq f(y)$) eminating from a vertex is $\mathbf{I}(f)$, the chance that the edge we pick will be a boundary edge is $\mathbf{I}(f)/n$.

Since $\mathbf{V}(f) = \sum_{k \neq 0} \mathbf{W}^k(f)$, and $\mathbf{I}(f) = \sum_{k \neq 0} k\mathbf{W}^k(f)$, it is fairly trivial to obtain the Poincare inequality $\mathbf{V}(f) \leqslant \mathbf{I}(f)$, which gives a strictly inequality unless $\mathbf{W}^{>2} f = 0$. For Boolean valued functions, the only such functions are the singular characters $\pm x^i$ and the constant functions. Geometrically, the poincare inequality gives an edge expansion bound for functions on the Hamming cube. Given a subset A of the n dimensional Hamming cube with m points, and if $\alpha = m/2^n$, then the 'characteristic function' $I_f : \{-1, 1\}^n \to \{-1, 1\}$ has variation $4\alpha(1 - \alpha)$, and therefore the fraction of edges in the cube which form a boundary edge is lower bounded by $4\alpha(1-\alpha)/n$. In particular, for the set A corresponding to $x^{[n]}$, where $\alpha = 1/2$, we see this bound is tight, as for $\alpha = 0$ and $\alpha = 1$. However, for other values of α must better bounds can be obtained.

Returning to our discussion of voting systems, an additional property to consider is the expected number of voters who agree with the outcome of the vote.

Theorem 2.3. *Let f be a voting rule for a 2-candidate election. If $w(x)$ is the number of x^i which agree with the voting choice, then*

$$\mathbf{E}(w) = \frac{n}{2} + \frac{1}{2} \sum \widehat{f}(i)$$

which is maximized by the majority functions.

Proof. We have

$$w(x) = \sum_{i=1}^{n} \frac{1 + f(x)x^i}{2}$$

Hence

$$\mathbf{E}(w(X)) = \frac{n}{2} + \frac{1}{2} \sum_{i=1}^{n} \mathbf{E}[f(x)x^i] = \frac{n}{2} + \frac{1}{2} \sum_{i=1}^{n} \widehat{f}(i)$$

We write

$$\sum_{i=1}^{n} \widehat{f}(i) = \mathbf{E}(f(X)(X^1 + X^2 + \cdots + X^n)) \leqslant \mathbf{E}(|X^1 + X^2 \cdots + X^n|)$$

and this inequality turns into an equality precisely when f has the property that $f(X) = \text{sgn}(\sum X^i)$, whenever $\sum X^i \neq 0$. □

Example. *Consider the function f choosen randomly from all Boolean valued functions on $\{-1,1\}^n$. We calculate*

$$\mathbf{E}[Inf_i(f)] = \mathbf{E}_X[\mathbf{P}(f(X) \neq f(X^{\oplus i}))] = \mathbf{E}_X[1/2] = 1/2$$

Hence $\mathbf{E}[\mathbf{I}(f)] = n/2$, so for an average function half of the voters agree with the outcome.

2.2 Noise and Stability

In realistic voting systems, the inputs to voting systems are rarely the most reliable. Some voters don't even show up to voting booths to announce their choice of candidate, and errors in the recording of data mean that the input might not even be accurate to the choices of voters. It is important for a voting rule to remain stable under these pertubations, so that the outcome of the modifed vote is likely to be the same as the outcome of the original vote, if we had perfect knowledge of all voting choices.

So our general rule with which we will define the **stability** of a voting rule f, is to find the probability that $f(x) \neq f(y)$, where y is obtained by a slight pertubation of x. We'll say two Boolean-valued random variables X and Y are p-correlated, for some $p \in [-1,1]$ if $\mathbf{E}[XY] = p$. This is equivalent to

$$\mathbf{P}(Y = X) = \frac{1+\rho}{2}$$

Given a random variable X, we can generate a random variable Y which is p-correlated to X by letting $Y = X$ with probability p, and otherwise choosing Y uniformly randomly. We write $Y \sim N_p(X)$ is Y is p-correlated to X. This is the sense with which we will need p-correlation, for p-correlation represents a certain value being randomly perturbed by a small quantity. Similarily, if X and Y are multidimensional Boolean-valued functions, we say X is p-correlated to Y if the coordinates of X and Y are independant, and each coordinate in X is p-correlated to the corresponding coordinate in Y. We can now define the stability with respect to p as

$$\text{Stab}_p(f) = \mathop{\mathbf{E}}_{Y \sim N_p(X)}[f(X)f(Y)]$$

where X is chosen uniformly. We of course find

$$\mathrm{Stab}_p(f) = 2\left(\mathop{\mathbf{P}}_{Y\sim N_p(X)} f(X) = f(Y)\right) - 1$$

Thus if the voting system is modified by some small random quantity p, the chance that this will affect the outcome of the vote is $[1 + \mathrm{Stab}_p(f)]/2$.

An alternate measure of the sensitivity of a function, which may be easier to reason with, is the probability that the vote is disrupted if we purposefully reverse each coordinate of X independently with a small probability δ, forming the random variable Y. Thus $\mathbf{P}(Y = X) = 1 - \delta$, hence Y is $1 - 2\delta$-correlated to X. We define the **noise sensitivity** based on this correlation to be

$$\mathbf{NS}_\delta(f) = \mathbf{P}[f(X) \neq f(Y)]$$

Since Y is $1 - 2\delta$ correlated with X, and by our previous formula, we find

$$\mathbf{NS}_\delta(f) = \frac{1 - \mathrm{Stab}_{1-2\delta}(f)}{2}$$

So that there is a linear correlation between the two measures of stability.

Example. *The constant functions ± 1 have stability 1, since there is no chance of changing the value of the functions. The dictator functions x^i satisfy $\mathbf{NS}_\delta(x^i) = \delta$, hence*

$$Stab_\rho(x^i) = 1 - 2\mathbf{NS}_{\frac{1-\rho}{2}}(x^i) = \rho$$

More generally, the stability of x^S is

$$Stab_\rho(x^S) = \mathbf{E}(X^S Y^S) = \prod_{i\in S} \mathbf{E}(X^i Y^i) = \prod_{i\in S}\left(\left(\frac{1+\rho}{2}\right) - \left(\frac{1-\rho}{2}\right)\right) = \rho^{|S|}$$

The noise stability of the majority functions has no convenient formula, but it does tend to a nice limit at the number of voters increases ad infinum. Later on, we will prove that for $\rho \in [-1, 1]$,

$$\lim_{\substack{n\to\infty \\ n\ odd}} Stab_\rho(Maj_n) = (2/\pi)\sin^{-1}(\rho)$$

Hence for $\delta \in [0,1]$,

$$\lim_{\substack{n\to\infty \\ n \text{ odd}}} \mathbf{NS}_\delta[Maj_n] = (1/\pi)\arccos(1-2\delta) = \frac{2\sqrt{\delta}}{\pi} + O(\delta^{3/2})$$

Hence the noise stability curves sharply near small and large values of ρ.

That noise stability is tightly connected to the Fourier coefficients of the function is one of the most powerful tools in Boolean harmonic analysis. As with the influence, we introduce an operator to provide us with a connection to the coefficients. We introduce the **noise operator** with parameter p as

$$(T_p f)(x) = \mathop{\mathbf{E}}_{X\sim N_p(x)} f(X)$$

which 'smoothes out' the function f by a certain parameter p. T_p is obviously linear, and

$$(T_p x^S) = \mathbf{E}[X^S] = \prod_{i\in S} \mathbf{E}[X^i] = \prod_{i\in S} \left(\left(\frac{1+\rho}{2}\right) x^i - \left(\frac{1-\rho}{2}\right) x^i \right) = \rho^{|S|} x^S$$

Hence the noise operator transforms the Fourier coeffients as

$$(T_p f)(x) = \sum p^{|S|} \widehat{f}(S) x^S$$

The connection between the noise operator and the stability is that

$$\begin{aligned} \mathrm{Stab}_p(f) &= \mathbf{E}[f(X)f(Y)] = \mathbf{E}[f(X)\mathbf{E}[f(Y)|X]] \\ &= \mathbf{E}[f(X)(T_p f)(X)] = \langle f, T_p f \rangle \end{aligned}$$

Thus stability is a quadratic form, and so we find

$$\mathrm{Stab}_p(f) = \sum_p \rho^{|S|} \widehat{f}(S)^2 = \sum_k \rho^k \mathbf{W}^k(f)$$

Correspondingly, we have

$$\begin{aligned} \mathbf{NS}_\delta(f) &= \frac{1-\mathrm{stab}_{1-2\delta}(f)}{2} \\ &= \frac{1-\mathbf{W}^0(f)}{2} - \sum_{k\neq 0} \frac{(1-2\delta)^k}{2} \mathbf{W}^k(f) \\ &= \frac{1}{2}\sum (1-(1-2\delta)^k)\mathbf{W}^k(f) \end{aligned}$$

Hence $\mathbf{NS}_\delta$ can also be represented as a quadratic form.

The operators T_ρ form a local one-parameter semigroup of the set of all operators on Boolean functions, because $T_\rho T_\mu = T_{\rho\mu}$. This essentially follows because if X is μ correlation to the constant $x \in \{-1, 1\}$, and Y is ρ correlated to X, then Y is $\rho\mu$ correlated to x, since

$$\begin{aligned}\mathbf{P}(Z = x) &= \left(\frac{1+\mu}{2}\right)\mathbf{P}(Y = X) + \left(\frac{1-\mu}{2}\right)\mathbf{P}(Y \neq X) \\ &= \left(\frac{1+\mu}{2}\right)\left(\frac{1+\rho}{2}\right) + \left(\frac{1-\mu}{2}\right)\left(\frac{1-\rho}{2}\right) \\ &= \frac{1+\rho\mu}{2}\end{aligned}$$

and this completes the argument. What's more, T_ρ is a contraction on $L^q\{-1,1\}^n$ for all $q \geqslant 1$, because by Jensen's inequality, since $x \mapsto x^q$ is convex,

$$\begin{aligned}\|T_\rho f\|_q^q &= \mathbf{E}[(T_\rho f)^q(X)] \\ &= \mathbf{E}[\mathbf{E}_{Y \sim N_\rho(X)}[f(Y)]^q] \\ &\leqslant \mathbf{E}[\mathbf{E}_{Y \sim N_\rho(X)}[f^q(Y)]]\end{aligned}$$

If X is chosen uniformly randomly, and $Y \sim N_\rho(X)$, then Y is also chosen uniformly randomly, since

$$\begin{aligned}\mathbf{P}(Y = 1) &= \mathbf{P}(Y = X)\mathbf{P}(X = 1) + \mathbf{P}(Y \neq X)\mathbf{P}(X = -1) \\ &= \frac{\mathbf{P}(Y = X) + \mathbf{P}(Y \neq X)}{2} = \frac{1}{2}\end{aligned}$$

Hence

$$\mathbf{E}[\mathbf{E}_{Y \sim N_\rho(X)}[f^q(Y)]] = \mathbf{E}[f^q(X)] = \|f\|_q^q$$

and we may obtain the inequality by taking q'th roots.

Using the Fourier representation, we can show that the dictator functions maximize stability among the unbiased Boolean-valued functions.

Theorem 2.4. *For $\rho \in (0,1)$, if f is an unbiased Boolean-valued function, then $Stab_p(f) \leqslant p$, with equality if and only if f is a dictator.*

Proof. We write

$$\mathrm{Stab}_p(f) = \sum p^{|S|}\widehat{f}(S)^2 \leqslant p\sum \widehat{f}(S)^2 = p$$

If this inequality is an equality, then $\mathbf{W}^{>1}(f) = 0$, and it then follows that f is either constant or a dictator function, and the constant functions are biased. □

Notice that $\mathrm{Stab}_p(f)$ is a polynomial in p, with non-negative coefficients. It follows that the stability of a function increases as p increases for positive p, and $\mathrm{Stab}_0(f) = 1$. What's more, we find that, looking at the coefficients, that

$$\frac{d\mathrm{Stab}_p(f)}{dp} = \sum_{S\neq\varnothing} |S|p^{|S|-1}\widehat{f}(S)^2$$

Hence the derivative at $p = 0$ is $\mathbf{W}^1(f)$, and the derivative at $p = 1$ is $\mathbf{I}(f)$.

We can incorporate stability into the concept of influence. For $p \in [0,1]$, define the p-**stable influence**

$$\mathrm{Inf}_i^{(p)}(f) = \mathrm{Stab}_p(D_i f) = \sum_{i\in S} \rho^{|S|-1}\widehat{f}(S)^2$$

The **total p-stable influence** is

$$\mathbf{I}^{(p)}(f) = \sum \mathrm{Inf}_i^{(p)}(f) = \sum_{i\in S} |S|\rho^{|S|-1}\widehat{f}(S)^2$$

In this form, we see that $\mathbf{I}^{(p)}(f) = d\mathrm{Stab}_p(f)/dp$, so that this p-stable influence measures the change in stability of a function with respect to p, but this isn't too useful, and there really isn't too much of a natural interpretation of the p-stable influences. Since $D_i f$ measures if f is pivotal at f, one intuitive idea of the p-stable influence is that it is a way of smoothening out how pivotal $D_i f$ is; Though $D_i f$ only measures the change in f over the i'th coordinate, the p-stable influence now incorporates a small change in the other coordinates as well. Regardless of their interpretation, they will be technically very useful later on.

Theorem 2.5. *Suppose that a function f satisfies $\mathbf{V}(f) \leqslant 1$. If $0 < \delta, \epsilon < 1$, then the number of indices i for which $Inf_i^{(1-\delta)}(f) \geqslant \varepsilon$ is less than or equal to $1/\delta\varepsilon$.*

Proof. Let J be the set of indices for which $\mathrm{Inf}_i^{(1-\delta)}(f) \geqslant \varepsilon$, so that we must prove $|J| \leqslant 1/\delta\varepsilon$. We obviously have $|J| \leqslant \mathbf{I}^{(1-\delta)}(f)/\varepsilon$, hence we need only prove $\mathbf{I}^{(1-\delta)}(f) \leqslant \frac{1}{\delta}$. Since

$$\begin{aligned}\mathbf{I}^{(1-\delta)}(f) &= \sum |S|(1-\delta)^{|S|-1}\widehat{f}(S)^2 \\ &\leqslant \max(|S|(1-\delta)^{|S|-1})\sum \widehat{f}(S)^2 = \max(|S|p^{|S|-1})\end{aligned}$$

Hence it suffices to prove that $k\delta(1-\delta)^{k-1} \leqslant 1$ for any $0 < \delta < 1$ and $k \geqslant 1$. Note that for $0 < x < 1$,

$$nx^n \leqslant \sum_{k=0}^{n} x^k \leqslant \sum_{k=0}^{\infty} x^k = 1/(1-x)$$

Our inequality follows by setting $x = 1 - \delta$. □

2.3 Arrow's Theorem

The majority system is perfect for a two-candidate voting system. It is monotone, symmetric, and maximizes the number of voters which agree with each choice. But for three-candidate preferential voting system, we have no information. Given a series of rankings of n candidates, we desire a way to decide upon a winner. In 1785, the French philosopher and mathematician Nicolas de Condorcet suggested breaking down individual voting preferences of voters into pairwise choices over candidates. Once these candidates are considered pairwise, we can then evaluate which candidate is the best over all comparisons. Thus we require a certain voting rule $f : \{-1,1\}^n \to \{-1,1\}$ to consider candidates pairwise, to consider such **Condorcet elections**. In this section, we consider only 3-candidate scenarios.

For instance, consider the Condorcet election over 3 candidates a, b, and c using the majority function as a voting rule. Consider three voters, with preferences $a > b > c$, $a > c > b$, and $b > c > a$. We see that b is favoured more often than c, whereas a is favoured more often than both b and c, hence a is the overall winner of the election. We say a is the **Condorcet winner**. Given a particular voter, we define the xy choice of the voter to be the element of $\{-1,1\}$, where 1 indicates the voter favours x, and -1 favours y. We can identify a particular choice of preference of

a voter by consider their ab preference, their bc preference, and their ca preference, which is an element of $\{-1,1\}^3$, and conversely, an element of $\{-1,1\}^3$ gives rise to a unique linear preference, provided that not all of the coordinates of the vector are equal, that is, the vector isn't $\{-1,-1,-1\}$ or $\{1,1,1\}$.

Unfortunately, Condorcet discovered that his election system may not lead to a definite winner. Consider the preferences $c > a > b$, $a > b > c$, and $b > c > a$, using the Condorcet method. Then a is preferred to b, b is preferred to c, and c is preferred to a, so there is no definite winner. In particular, if we consider the winner of each election as a tuple of ab, bc, and ca preferences in $\{-1,1\}^3$, then we can reconstruct a Condorcet winner if and only if the coordinates of the preference vector aren't $\{-1,-1,-1\}$ or $\{1,1,1\}$. Thus the majority rule appears to be deficient in a preferential election.

It was Kenneth Arrow in the 1950s who found that there is always a chance that a Condorcet winner does not occur with *any* voting rule to decide upon a pariwise Condorcet winner, except for a particullary unattractive option: the dictator function. In 2002, Kalai gave a new proof of this theorem using the tools of Fourier analysis. As should be expected from our analysis, it looks at the properties of the 'not all equal' function NAE : $\{-1,1\}^3 \to \{0,1\}$. Instead of looking at a particular element of the domain of the function to determine whether a Condorcet winner is not always possible, Kalai instead computes the *probability* of a Condorcet winner of a voting rule, where the preferences are chosen over all possibilities.

Lemma 2.6. *The probability of a Condorcet winner in a Condorcet election using the pairwise voting rule $f : \{-1,1\}^n \to \{-1,1\}$ is precisely*

$$\frac{3}{4}[1 - Stab_{-1/3}(f)]$$

Proof. Let $x, y, z \in \{-1,1\}^n$ be chosen such that (x_i, y_i, z_i) gives the (ab, bc, ca) preferences of a voter i. The x, y, z are chosen uniformly, in such a way that each (x_i, y_i, z_i) is chosen independently, and uniformly over the 6 tuples which satisfy the not all equal predicate. There is a Condorcet winner if and only if $\mathrm{NAE}(f(x), f(y), f(z)) = 1$, hence

$$\mathbf{P}(f \text{ gives a Condorcet winner}) = \mathbf{E}[\mathrm{NAE}(f(X), f(Y), f(Z))]$$

and since NAE has the Fourier expansion

$$\mathrm{NAE}(x,y,z) = \frac{3 - xy - xz - yz}{4}$$

Hence

$$\mathbf{E}[\mathrm{NAE}(f(X),f(Y),f(Z))] = \frac{3}{4} - \frac{\mathbf{E}(f(X)f(Y)) + \mathbf{E}(f(Y)f(Z)) + \mathbf{E}(f(Z)f(X))}{4}$$

Finally, note that each coordinate of X,Y,Z is independant, and

$$\mathbf{E}[XY] = \mathbf{E}[YZ] = \mathbf{E}[ZX] = 2/6 - 4/6 = -1/3$$

Thus (X,Y), (Y,Z), and (Z,X) are $-1/3$ correlated, hence

$$\mathbf{E}[\mathrm{NAE}(f(X),f(Y),f(Z))] = (3/4)(1 - \mathrm{stab}_{-1/3}(f))$$

and this gives the required formula. □

Example. *Using the majority function Maj_n, we find that the probability of a Condorcet winner tends to*

$$\frac{3}{4}(1 - (2/\pi)\sin^{-1}(-1/3)) = \frac{3}{2\pi}\cos^{-1}(-1/3)$$

thus there is approximately a 91.2% *chance of a Condorcet winner occuring. This is Guilbaud's formula.*

Theorem 2.7 (Kalai)**.** *The only voting rule $f : \{-1,1\}^n \to \{-1,1\}$ which always gives a Condorcet winner is the dictator functions x^i, or its negation.*

Proof. If f always gives a Condorcet winner, the chance of a winner is 1, hence rearranging the formula just derived, we find $\mathrm{stab}_{-1/3}(f) = -1/3$. But then

$$\sum(-1/3)^k \mathbf{W}^k(f) = -1/3$$

Since $(-1/3)^k > -1/3$ for $k > 1$, we find

$$\sum(-1/3)^k \mathbf{W}^k(f) \geqslant -1/3 \sum \mathbf{W}^k(f) = -1/3$$

and this inequality becomes an equality if and only if $\mathbf{W}^{>1}(f) = 0$, which we have verified implies that f is constant, a dictator function, or it's negation. Assuming f is unanimous, we find that f must be the dictator function. □

We might wonder which voting rule gives us the largest chance of a Condorcet winner. It turns out that we can only obtain a slightly more general result than for the majority function.

Lemma 2.8. *If $f : \{-1,1\} \to \{-1,1\}$ has all $\widehat{f}(i)$ equal, then*

$$\mathbf{W}^1(f) \leqslant 2/\pi + O(1/n)$$

Proof. Since $n\widehat{f}(i) = \sum \widehat{f}(i) \leqslant \sqrt{2n/\pi} + O(\sqrt{1/n})$ (the majority function maximizes $\sum \widehat{f}(i)$), we find

$$\mathbf{W}^1(f) = n\widehat{f}(i)^2 = \frac{1}{n}(n\widehat{f}(i))^2 \leqslant 2/\pi + O(1/n)$$

Hence the inequality holds. □

Theorem 2.9. *In a 3-candidate Condorcet election with n voters, the probability of a Condorcet winner is at most* $(7/9) + (2/9)\mathbf{W}^1(f)$

Proof. Given any $f : \{-1,1\}^n \to \{-1,1\}$, the probability of a Condorcet has a Fourier expansion

$$\begin{aligned}
\frac{3}{4} - \frac{3}{4}\mathrm{Stab}_{-1/3}(f) &= \frac{3}{4} - \frac{3}{4}\sum(-1/3)^k \mathbf{W}^k(f) \\
&= \left(\frac{3}{4} - \mathbf{E}[f]\right) + \frac{\mathbf{W}^1(f)}{4} - \frac{\mathbf{W}^2(f)}{12} + \frac{\mathbf{W}^3(f)}{36} - \dots \\
&\leqslant (3/4) + \frac{\mathbf{W}^1(f)}{4} + \frac{1}{36}\left(\sum_{k=2}^{n} \mathbf{W}^k(f)\right) \\
&\leqslant (3/4) + \frac{\mathbf{W}^1(f)}{4} + \frac{1 - \mathbf{W}^1(f)}{36} = 7/9 + 2/9\mathbf{W}^1(f)
\end{aligned}$$

□

Combining these two results, we can conclude that in any voting system with all $\widehat{f}(i)$ equal (which is certainly true even in a transitive symmetric system), the chance of a Condorcet winner is

$$(7/9) + (2/9)\mathbf{W}^1(f) \leqslant (7/9) + (4/9\pi) + O(1/n)$$

And this is approximately 91.9%. Later on, we will be able to show that this bound holds given a much weaker hypothesis, and what's more, we will show that the majority function actually maximizes the probability.

The advantage of Kalai's approach is that we can use it to obtain a much more robust result, ala linearity testing.

Theorem 2.10. *If the probability of a Condorcet election of* $1-\varepsilon$*, then* f *is* $O(\varepsilon)$ *close to being a dictator, or the negation of a dictator.*

Proof. Using the bound we just used, we find $1-\varepsilon \leqslant 7/9+2/9\mathbf{W}^1(f)$, hence

$$1-(9/2)\varepsilon \leqslant \mathbf{W}^1(f)$$

and then the result follows from the Friedgut Kalai Naor theorem. □

Theorem 2.11 (Friedgut-Kalai-Naor theorem)**.** *If* f *is a Boolean valued function, and* $\mathbf{W}^1(f) \geqslant 1-\delta$*, then* f *is* $O(\delta)$ *close to* $\pm x^i$ *for some index* i*.*

Chapter 3

Spectral Complexity

In this chapter, we discuss ways we can measure the 'complexity' of a Boolean function, which take the form of various definitions relating to the Fourier coefficients of the function, and the shortest way we can describe the function's output. This discussion has immediate applications to learning theory, where we desire functions which can easily be learned (the simplest functions), and cryptography, where we desire functions which cannot be easily deciphered.

3.1 Spectral Concentration

One way for a function to be 'simple' is for its Fourier coeffients to be concentrated on coefficients of small degree. Call a Boolean function f ε-**concentrated** on degree n if $\mathbf{W}^{\geqslant n}(f) \leqslant \varepsilon$. For Boolean valued functions, this is equivalent to saying that $\mathbf{P}(|S| > n) \leqslant \varepsilon$, where S has the induced spectral distribution. The total influence provides an initial estimate of the spectral concentration.

Theorem 3.1. *A Boolean f is ε-concentrated on degree n, for $n \geqslant \mathbf{I}(f)/\varepsilon$.*

Proof. If we write

$$\mathbf{I}(f) = \sum_k k\mathbf{W}^k(f) \leqslant \varepsilon n$$

then $n\mathbf{W}^{\geqslant n}(f) \leqslant \sum_k k\mathbf{W}^k(f)$, and combining these inequalities completes the proof. For boolean valued functions, this is essentially Markov's inequality applied to the spectral distribution. □

Example. *The tribes function $Tribes_{w,2^w}$ has total influence $O(\log(w2^w)) = O(w)$, hence for any constant c, the tribes function is c-concentrated on degree up to $O(w)$.*

Another estimate of the spectral concentration is obtained from the noise stability.

Theorem 3.2. *For any Boolean-valued function f, and $0 < \delta \leqslant 1/2$, the Fourier spectrum of f is ε-concentrated on degree $n \geqslant 1/\delta$ for*

$$\varepsilon = \frac{2}{1-e^{-2}}\mathbf{NS}_\delta(f) \leqslant 3\mathbf{NS}_\delta(f)$$

Proof. Using the spectral formula for the noise sensitivity, if S takes on the spectral distribution, then applying the Markov inequality, we find

$$\begin{aligned} 2\mathbf{NS}_\delta(f) &= \mathbf{E}[1-(1-2\delta)^{|S|}] \\ &\geqslant (1-(1-2\delta)^{1/\delta})\mathbf{P}(|S| \geqslant 1/\delta) \\ &\geqslant (1-e^{-2})\mathbf{P}(|S| \geqslant n) \end{aligned}$$

where we used that $1-(1-2\delta)^k$ is a non-negative, non-decreasing function of k, and $(1-2\delta)^{1/\delta} \leqslant e^{-2}$. □

Example. *For δ sufficiently small, and n sufficiently large, the noise sensitivity of Maj_n with parameter δ is bounded by $\sqrt{\delta}$. Hence Maj_n is $3\sqrt{\delta}$ concentrated on degree m for $m \geqslant 1/\delta$.*

We can push ε-concentration all the way down to 0-concentration on degree n, which is the same as saying the Fourier expansion of the function in question has degree less than or equal to n. For these function, we can actually control how many coordinates of the function are useful to the function.

Lemma 3.3. *If f is a real-valued Boolean function with $f \neq 0$, and the degree of the Fourier expansion is less than or equal to m, then $\mathbf{P}(f(X) \neq 0) \geqslant 2^{-m}$.*

Proof. We prove this by induction on the number of coordinates of f. If $f(x) = a$ is a constant function, and $a \neq 0$, then $\mathbf{P}(f(X) \neq 0) = 1 \geqslant 2^{-0}$. If $f(x) = a + bx$, and $b \neq 0$, then either $f(1)$ or $f(-1)$ are non-zero, so the lemma is satisfied here. In general, if we write

$$f(x) = \sum \widehat{f}(S)x^S$$

and then we define $n-1$ dimensional functions $f_1(x) = f(x,1)$, and $f_{-1}(x) = f(x,-1)$, then

$$f_1(x) = \sum_{n \notin S} \left(\widehat{f}(S) + \widehat{f}(S \cup \{i\}) \right) x^S \qquad f_{-1}(x) = \sum_{n \notin S} \left(\widehat{f}(S) - \widehat{f}(S \cup \{i\}) \right) x^S$$

Then either f_1 has degree $m-1$, or f_{-1} has degree $m-1$, and it follows by induction that

$$\mathbf{P}(f(X) \neq 0) = \frac{\mathbf{P}(f_1(X) \neq 0) + \mathbf{P}(f_{-1}(X) \neq 0)}{2} \geqslant \frac{1}{2^{(m-1)}} \frac{1}{2} = 2^{-m}$$

and by induction, the bound holds for all functions f. □

Using this lemma, since $\mathrm{Inf}_i(f) = \mathbf{P}((D_i f)(X) \neq 0)$, and if $\deg(f) \leqslant k$, then $\deg(D_i f) \leqslant k-1$, so either $\mathrm{Inf}_i(f) = 0$, or $\mathrm{Inf}_i(f) \geqslant 2^{1-k}$.

Theorem 3.4. *If* $\deg(f) \leqslant k$, *then* f *is a* $k2^{k-1}$ *junta.*

Proof. Because $\mathrm{Inf}_i(f) = 0$ or $\mathrm{Inf}_i(f) \geqslant 2^{1-k}$, the number of coordinates with non-zero influence on f is at most $\mathbf{I}(f)/2^{1-k}$, and those coordinates with non-zero influence are irrelevant to the functions definition. Since

$$\mathbf{I}(f) = \sum_{m \leqslant k} m \mathbf{W}^m(f) \leqslant k(\mathbf{W}^1(f) + \cdots + \mathbf{W}^k(f)) = k$$

hence $\mathbf{I}(f)/2^{1-k} \leqslant k2^{k-1}$, and so f is a $k2^{k-1}$ junta. □

The Friedgut Kalai Naor theorem is essentially a more robust version of this theorem, in the case where $k = 1$.

3.2 Decision Trees

Another classification of 'simple' Boolean functions occurs by analyzing the decision trees which represent the function. A **decision tree** for a function $f : \mathbf{F}_2^n \to \mathbf{R}$ is a representation of f by a binary tree in which the internal nodes of the tree are labelled by variables, edges labelled 0 or 1, and leaves by real numbers, such that $f(x)$ can be obtained by starting at the head of the tree, then proceeding down based on the input and the labels of the tree. The **size** of the tree is the number of leaves, and

the **depth** is the maximum possible length of a branch from root to leaf. The depth counts the maximum number of questions we need to ask to determine the output of a function f.

Essentially, a decision tree subdivides $\mathbf{F}_2^n$ into cubes upon which the represented function is constant. These cubes can be identified as the set of inputs which follows the same path P, and we shall denote the cube by C_P. It follows that

$$f(x) = \sum_{\text{paths } P \text{ of } T} f(P)\mathbf{I}(x \in C_P)$$

This essentially tells us that low-depth decision tress should have low spectral complexity, because the length of paths bound the degree of the Fourier coefficients. Introducing terminology which will soon become more important, we define the **sparsity** of a function f, denoted sparsity(f), to be the number of elements of its domain which are non-zero. We let the **spectral sparcity** of f be sparsity$(\widehat{f})$, it is the number of subsets $S \subset [n]$ such that $\widehat{f}(S) \neq 0$. Also define f to be ε-**granular** if $f(x)$ is a multiple of ε for any x in the domain.

Theorem 3.5. *If f can be computed by a decision tree of size s and depth k, then*

(1) $\deg(f) \leqslant k$.

(2) *sparsity*$(\widehat{f}) \leqslant s2^k \leqslant 4^k$

(3) $\|\widehat{f}\|_1 \leqslant s\|f\|_\infty \leqslant 2^k\|f\|_\infty$.

(4) *$\widehat{f}$ is 2^{-k}-granular, assuming f is integer valued.*

Proof. If a path P has depth m, querying if $x_{i_1} = a_{i_1}, x_{i_2} = a_{i_2}, \ldots, x_{i_m} = a_{i_m}$, then

$$\begin{aligned}\mathbf{I}(x \in C_P) &= \mathbf{I}(x_{i_1} = a_1, \ldots, x_{i_m} = a_m) \\ &= \frac{1}{2^m}\prod_{j=1}^{m}\left(1 + a_j x_{i_j}\right) = \frac{1}{2^m}\sum_{S \subset \{i_1,\ldots,i_m\}} a^S x^S\end{aligned}$$

which we see has a Fourier expansion with degree less than or equal to m. It follows that if each path determining f has degree bounded by k, then

the degree of the Fourier coefficients of f are bounded by k, because of the sum formula we calculated above, hence (1) follows. We also see that each path of length m adds at most 2^m Fourier coefficients to the equation, hence (2) follows (also, we see that similar branches share many of the same Fourier coefficients, hence this bound can often be tightened). What's more,

$$f(x) = \sum_{\text{paths } P \text{ of } T} \frac{f(P)}{2^{l(P)}} \sum_{S \subset \{i_1^P, \ldots, i_{l(P)}^P\}} a_P^S x^S$$

Thus

$$\|\widehat{f}\|_1 \leqslant \sum_{\text{paths } P \text{ of } T} \sum_{S \subset \{i_1^P, \ldots, i_{l(P)}^P\}} \frac{|f(P)|}{2^{l(P)}} \leqslant s\|f\|_\infty$$

hence (3) follows. Since $a_P^S \in \{-1, 1\}$, if $f(P) \in \mathbf{Z}$ then $f(P)a_P^S/2^{l(P)}$ is 2^{-k} granular, hence, summing up, we find all the coefficients of f are. □

Theorem 3.6. *If $f : \{-1,1\}^n \to \{-1,1\}$ is computable by a decision tree of size s, and $0 < \varepsilon \leqslant 1$, then the spectrum of f is ε concentrated on degree $m \geqslant \log(s/\varepsilon)$.*

Proof. Let T be the decision tree of size s, and consider the tree T_0 obtained by truncating each branch of length greater than or equal to $\log(s/\varepsilon)$. New leaves created can be chosen arbitrarily. Then the function g obtained from T_0 is ε-close to f, because g only differs from f on the paths of length $l \geqslant \log(s/\varepsilon)$, and since each query the decision tree makes subdivides the domain in two, changes only $2^{-l} \leqslant 2^{-\log(s/\varepsilon)} = \varepsilon/s$ elements of the domain, and since we only have s paths, we have only changed ε elements of the domain. Since if $m \geqslant \log(s/\varepsilon)$, $\widehat{g}(S)$ vanishes for $|S| \geqslant m$, hence

$$\mathbf{W}^{\geqslant m}(f) \leqslant \|\widehat{g} - \widehat{f}\|_2^2 = \|g - f\|_2^2 \leqslant 4\varepsilon$$

COME UP WITH TIGHTER BOUND LATER. □

3.3 Computational Learning Theory

Computational learning theory attempts to solve the following problem, motivated from statistics. Given a class of functions $\mathcal{C}$, we attempt to determine an unknown **target function** $f \in \mathcal{C}$ with limited access to that

function. We shall assume, of course, that our Boolean-valued functions. The two models of obtaining data to learn the target function are:

- **random examples**, where we draw random samples $(X, f(X))$, where X is uniformly random on the domain.
- **queries**, where we can request a specific value x for any $x \in \{-1, 1\}^n$.

The query method is at least as powerful as the random example method, because we can always randomly choose x ourselves. After a certain number of steps, we are required to choose a **hypothesis function** $h \in \mathcal{C}$. We say that an algorithm **learns** $\mathcal{C}$ **with error** ε if for any $f \in \mathcal{C}$, the algorithm outputs a hypothesis function which is ε-close to f with high probability. The particular accuracy probability baseline is irrelevant, since we can always adjust the baseline by introducing a polynomial number of steps.

As should be intuitive, the more 'simple' the class $\mathcal{C}$ is, the faster it should be to recognize a hypothesis function. If $\mathcal{C}$ is suitably complex, then exponential running time may be necessary. Generalizing our definition of concentration of spectrum, we say f is ε concentrated on $\mathcal{F} \subset 2^{[n]}$ if

$$\sum_{S \notin \mathcal{F}} \widehat{f}(S)^2 \leqslant \varepsilon$$

The main result is that learning a function is equivalent to learning it's Walsh Fourier spectrum, so that functions concentrated on a small portion of the Fourier space are simpler to learn.

Theorem 3.7. *Given access to random samples to a Boolean-valued function, there is a randomized algorithm which take $S \subset [n]$ as input, $0 \leqslant \delta, \varepsilon \leqslant 1/2$, and outputs an estimate $\widetilde{f}(S)$ for $\widehat{f}(S)$ such that*

$$\mathbf{P}(|\widetilde{f}(S) - \widehat{f}(S)| > \varepsilon) \leqslant \delta$$

in $\log(1/\delta) \cdot Poly(n, 1/\varepsilon)$ time.

Proof. We have $\widehat{f}(S) = \mathbf{E}[f(X)X^S]$. Given random samples

$$(X_1, f(X_1)), \ldots, (X_m, f(X_m))$$

we can compute $f(X_k)X_k^S$, and then estimate $\mathbf{E}[f(X)X^S]$. A standard application of the Chernoff bound shows that $O(\log(1/\delta)/\varepsilon^2)$ examples are sufficient to obtain an estimate within $\pm\varepsilon$ with probability at least $1 - \delta$. □

Theorem 3.8. *If f is a Boolean-valued function, g is a real-valued Boolean function, and $\|f-g\|_2^2 \leqslant \varepsilon$. Let $h(x) = sgn(g(x))$, with $sgn(0)$ chosen arbitrarily. Then $d(f,h) \leqslant \varepsilon$.*

Proof. Since $|f(x)-g(x)|^2 \geqslant 1$ when $f(x) \neq \operatorname{sgn}(g(x))$,

$$d(f,h) = \mathbf{P}(f(X) \neq h(X)) \leqslant \mathbf{E}[|f(X)-g(X)|^2] = \|f-g\|_2^2$$

Thus we can accurately 'discretize' a real-valued estimate of a Boolean-valued function. □

These theorems allow us to show that estimating Fourier coeficients suffices to estimate the function.

Theorem 3.9. *If an algorithm has random sample access to a Boolean function f, and can identify a family $\mathcal{F}$ upon which f's Fourier spectrum is ε concentrated, then in poly$(|\mathcal{F}|, n, 1/c)$ time, we can with high probability produce a hypothesis function $\varepsilon + c$ close to f.*

Proof. For each $S \in \mathcal{F}$, we produce an estimate $\widetilde{f}(S)$ such that

$$|\widetilde{f}(S) - \widehat{f}(S)| \leqslant \delta$$

except with probability bounded by ρ, in $\log(1/\rho) \cdot \operatorname{poly}(n, 1/\delta)$ time. Applying a union bound, we find all inequalities hold except with probability $\delta|\mathcal{F}|$. Finally, we form the function $g(x) = \sum \tilde{f}(S)x^S$, and then output $h(x) = \operatorname{sgn}(g(x))$. Now by the last lemma, it suffices to bound the L_2 norm of f and g, and

$$\begin{aligned}
\|f-g\|_2^2 &= \sum \left(\widehat{f}(S) - \widehat{g}(S)\right)^2 \\
&= \sum_{S \in \mathcal{F}} \left(\widehat{f}(S) - \tilde{f}(S)\right)^2 + \sum_{S \notin \mathcal{F}} \widehat{f}(S)^2 \\
&= |\mathcal{F}|\delta^2 + \varepsilon
\end{aligned}$$

Hence h is $|\mathcal{F}|\delta^2 + \varepsilon$ close to f, and we have computed h in $\log(1/\rho) \cdot \operatorname{poly}(n, 1/\delta)$ time. If we let $\delta = \sqrt{c/|\mathcal{F}|}$, we find h is $c + \varepsilon$ close to f, in time $\log(1/\rho) \cdot \operatorname{poly}(n, |\mathcal{F}|, 1/c)$. □

If we assume that we are learning over a simple concept class to begin with, this theorem essentially provides all the information we need to know in order to learn efficiently over this concept class.

Example. *If our concept class consists of elements in $\mathcal{C}$, which only contains functions of degree $\leqslant k$, then $\mathcal{C}$ is 0-concentrated on the Fourier coefficients of degree k, which contains*

$$|\mathcal{F}| \leqslant \sum_{i=0}^{k} \binom{n}{k} = O(n^k)$$

elements. Thus in $\log(1/\rho) \cdot poly(n^k, 1/\varepsilon)$, we can learn an element of $\mathcal{C}$ with probability $1 - 1/\rho$ up to an accuracy ε.

Example. *If $\mathcal{C}$ consists of elements f with $\mathbf{I}(f) \leqslant t$, then each element is ε concentrated on degree past t/ε, thus we can learn elements of this set in $\log(1/\rho) \cdot poly(n^{t/\varepsilon}, 1/c)$ with error probability ρ, and error $\varepsilon + c$.*

Example. *If $\mathcal{C}$ consists solely of monotone functions, then the total influence of the functions is bounded by $\sqrt{2n/\pi} + O(1/\sqrt{n}) = O(\sqrt{n})$, hence we can learn elements of $\mathcal{C}$ in $\log(1/\rho) \cdot poly(n^{O(\sqrt{n})/\varepsilon}, 1/c)$ with error probability ρ, and error $\varepsilon + c$.*

Example. *If a class $\mathcal{C}$ consists of functions f such that $\mathbf{NS}_\delta(f) \leqslant 3\varepsilon$, for $\delta \in (0, 1/2]$, then f is ε concentrated on degree $1/\delta$, hence $\mathcal{C}$ is learnable with error $\varepsilon + c$ and error probability ρ in time $\log(1/\rho) \cdot poly(n^{1/\delta}, 1/c)$*

Example. *If a class $\mathcal{C}$ consists of functions f which can be represented by a decision tree with size bounded by s, then the spectrum of f is ε-concentrated on degree $\log(s/\varepsilon)$, hence $\mathcal{C}$ is learnable with error $\varepsilon + c$ and error probability ρ in time $\log(1/\rho) \cdot poly(n^{\log(s/\varepsilon)}, 1/c) = \log(1/\rho) \cdot poly(s/\varepsilon, 1/c)$.*

3.4 Goldreich-Levin Theorem

We have now reduced our problem to identifying a family $\mathcal{F}$ upon which f is concentrated. One such method is provided by the Goldreich-Levin algorithm, which can find $\mathcal{F}$ assuming query access to f. In order for the algorithm to terminate in polynomial time, we require that there is a set

$\mathcal{F}$ which exists and is small, yet the algorihtm will find a concentration family $\mathcal{F}$ given arbitrary input.

First, we consider the algorithm's origin in crytography. The goal of this field of study is to find 'encryption functions' f such that if x is a message, then $f(x)$ is easy to compute, but it is very difficult to compute x given $f(x)$. The strength of the encryption f is the difficulty in how it can be inverted. The Goldreich Levin theorem is a tool for building strong encryption schemes from weak schemes. Essentially, this breaks down into finding linear function slightly correlated to some fixed function f, given query access to the function.

The Goldreich Levin theorem assumes query access to a Boolean-valued function $f : \{-1,1\}^n \to \{-1,1\}$, and some fixed $\tau > 0$. Then in time $\text{poly}(n, 1/\tau)$, the algorithm outputs a family $\mathcal{F}$ of subsets of $[n]$ such that

1. If $|\widehat{f}(S)| \geqslant \tau$, $S \in \mathcal{F}$.
2. If $S \in \mathcal{F}$, $|\widehat{f}(S)| \geqslant \tau/2$.

Given a particular S, we know we can compute the estimated Fourier coefficients $\tilde{f}(S)$ to an arbitrary degree of precision with a high degree of accuracy. The problem is that if we did this for all S, then our algorithm wouldn't terminate in polynomial time. The Goldreich-Levin algorithm uses a divide-and-conquer strategy to measure the Fourier weight of the function over various collections of sets.

Given $f : \{-1,1\}^n \to \mathbf{R}$, it will be more natural to think of the function as $f : \{-1,1\}^{[n]} \to \mathbf{R}$, i.e. the function which takes an input $x : [n] \to \{-1,1\}$, and outputs a number $f(x)$. We can consider the Fourier expansion of an arbitrary $f : \{-1,1\}^A \to \mathbf{R}$, where A is an arbitrary index set, and then the characters will take the form x^S for $S \subset A$, and we can consider the Fourier coefficients. Let (J, J^c) be a partition of $[n]$. Then for $z \in \{-1,1\}^J$, we can define $f_{J|z} : \{-1,1\}^{J^c} \to \mathbf{R}$ to be the function obtained by fixing all indices of J with the values z. It follows that if $S \subset J^c$, then

$$\widehat{f_{J|z}}(S) = \sum_{T \subset J} \widehat{f}(S \cup T) z^T$$

Then if we pick z randomly over all choices, we find

$$\mathbf{E}_Z[\widehat{f_{J|Z}}(S)] = \sum_{T \subset J} \widehat{f}(S \cup T) \mathbf{E}[Z^T] = \widehat{f}(S)$$

If we define $F_{S|J^c} : \{-1,1\}^{J^c} \to \mathbf{R}$, for $S \subset J$, defined by

$$(F_{S|J^c} f)(z) = \widehat{f_{J|z}}(S) = \sum_{T \subset J^c} \widehat{f}(T \cup S) z^T$$

Then this gives a Fourier expansion for the funciton $(F_{S|J^c} f)$ and

$$\mathbf{E}_Z[\widehat{f}_{J|Z}(S)^2] = \mathbf{E}[(F_{S|J^c} f)(Z)^2] = \sum_{T \subset J^c} \widehat{f}(T \cup S)^2$$

Given f and $S \subset J \subset [n]$, let

$$\mathbf{W}^{S|J^c}(f) = \sum_{T \subset J^c} \widehat{f}(S \cup T)^2$$

which is the weight over irrelevant indices. A crucial tool of Golreich-Levin is that this is the expected value of $\widehat{f}_{J|Z}(S)^2$.

Theorem 3.10. *For any $S \subset J \subset [n]$, an algorithm with query access to $f : \{-1,1\}^n \to \{-1,1\}$ can compute an estimate of $\mathbf{W}^{S|J^c}(f)$ that is accurate to with $\pm\varepsilon$, except with probability at most δ, in poly$(n, 1/\varepsilon) \cdot \log(1/\delta$ time.*

Proof. We can write

$$\begin{aligned}\mathbf{W}^{S|J^c}(f) = \mathbf{E}_Z[\widehat{f_{J|Z}}(S)^2] &= \mathbf{E}_Z\left[\mathbf{E}_Y\left[f_{J|Z}(Y) Y^S\right]^2\right] \\ &= \mathbf{E}_Z\left[\mathbf{E}_{Y_0,Y_1}\left[f_{J|Z}(Y_0) Y_0^S f_{J|Z}(Y_1) Y_1^s\right]\right]\end{aligned}$$

using queries to f, we can sample the inside of this equation, and a Chernoff bound shows that $O(\log(1/\delta)/\varepsilon^2)$ samples are enough for the estimate to have accuracy ε with confidence $1 - \delta$. □

We can now describe the Goldreich Levin algorithm. Initially, all subsets of $[n]$ are put in a single 'bucket'. We then repeat the following process:

- Select any bucket B with 2^m sets in it.
- Split B into two buckets B_1 and B_2 of 2^{m-1} sets.
- Estimate $\sum_{U \in B_1} \widehat{f}(U)^2$ and $\sum_{U \in B_2} \widehat{f}(U)^2$.

- Discard B_1 and B_2 if the weight estimate is less than or equal to $\tau^2/2$.

The algorithm stops when each bucket contains a single item. Note that we never discard a set S with $|\widehat{f}(S)| \geqslant \tau$, because $\tau^2 \geqslant \tau^2/2$. Furthermore, if a bucket containing two elements splits into two buckets containing a single element, then if S is undiscarded in this process, we must have $|\widehat{f}(S)| \geqslant \tau/2$, else $|\widehat{f}(S)|^2 \leqslant \tau^2/4 \leqslant \tau^2/2$. Note that we need only calculate the estimates up to $\pm\tau^2/4$ for the algorithm to be correct, so we have some wriggle room to work with.

Now we need to determine how fast it takes for the algorithm to terminate. Every bucket that isn't discarded has weight $\tau^2/4$, so Parseval's theorem tells us that there is never more than $4/\tau^2$ active buckets. Each bucket can only be split n times, hence the algorithm repeats its main loop $4n/\tau^2$ times. If the buckets can be accurately weighed and maintained in $\mathrm{poly}(n, 1/\tau)$ time, the overall running time will be $\mathrm{poly}(n, 1/\tau)$.

Finally we describe how to weight the buckets. Let

$$B_{k,S} = \{S \cup T : T \subset \{k+1, k+2, \ldots, n\}\}$$

Then $|B_{k,S}| = 2^{n-k}$, the initial bucket is $B_{0,\varnothing}$, and any bucket $B_{k,S}$ splits into $B_{k+1,S}$ and $B_{k+1,S\cup\{k\}}$. The weight of $B_{k,S}$ is $\mathbf{W}^{S|\{k+1,\ldots,n\}}(\mathrm{f})$, and can be measured to accuracy $\pm\tau^2/4$ with confidence $1-\delta$ in $\mathrm{poly}(n, 1/\tau) \cdot \log(1/\delta)$. The algorithm needs at most $8n/\tau^2$ weighings, hence by setting $\delta = \tau^2/(80n)$, we can ensure all weights are accurate with high probability, and the algorithm is $\mathrm{poly}(n, 1/\tau)$.

Chapter 4

Property Testing

We now study property testing, probabilistically checkable proofs, and constraint satisfication problems. All our work is centered around *dictatorship testing* – checking if an arbitrary boolean-valued function is a dictator. It turns out that being able to test if a certain function is a dictator is sufficient to test *any* property of boolean-valued functions, provided that the right 'proof' is provided of the property.

The field of property testing asks a simple problem. Given a collection $\mathcal{C}$ of n-bit Boolean-valued functions, and a function $f : \{-1,1\}^n \to \{-1,1\}$, how easy is to determine if $f \in \mathcal{C}$ using random examples or queries of the function. An **r-query testing alogirhtm** for $\mathcal{C}$ is an algorithm which randomly chooses r elements $x_1,\ldots,x_n \in \{0,1\}^n$, queries $f(x_1),\ldots,f(x_n)$, and decides deterministically whether $f \in \mathcal{C}$. A **local tester** with rejection rate $\lambda > 0$ is an r-query tester which always accepts f if $f \in \mathcal{C}$, and if $d(f,\mathcal{C}) > \varepsilon$, then the tester rejects f with probability exceeding $\lambda\varepsilon$.

Example. *The BLR test is a 3-query local tester with rejection rate 1 for the class of linear functions. Technically, the BLR test is really a family of testers for each dimension of Boolean function.*

Bibliography

[1] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, M. Sudan. *Linearity Testing in Characteristic Two.*