The Harmonic Analysis of Boolean Functions

Jacob Denson

September 14, 2022

Table Of Contents

Chapter 1

Introduction to Boolean Analysis

1.1 Introduction

These notes dicuss the theory of discrete harmonic analysis. The philosophy of discrete harmonic analysis is that a well-crafted 'binary encoding' of a set X encodes many of its interesting combinatorial properties. From the viewpoint of harmonic analysis, the abelian group structure on the set $\{0,1\}^n$ is the main object of study. A one-to-one correspondence between X and $\{0,1\}^n$ induces an abelian group structure on X, and we obtain powerful results about properties of X through the Fourier transform of the abelian group, provided that these properties play well with the pointwise exclusive or operation on $\{0,1\}$, which are simultaneously diagonalized by the Fourier transform.

Example. *The* Hamming distance $\Delta(x,y)$ *between two bit strings* $x,y \in \{0,1\}^n$ *is the number of coordinates where the two strings differ in value. For any big string* z*, we have*

$$\Delta(x+z,y+z) = \Delta(x,y)$$

so the Fourier transform will likely tell us information about the Hamming distance. This is especially true if we switch to studying the Hamming distance between two Boolean functions

$$f : \{0,1\}^n \to \{0,1\} \quad \textit{and} \quad g : \{0,1\}^n \to \{0,1\},$$

which we can identify as bit strings of length 2^n*, since then*

$$\Delta(f,g) = \#\{x \in \{0,1\}^n : f(x) \neq f(y)\}$$

can be identified with the L^1 distance between the two functions f and g with respect to the counting measure on $\{0,1\}^n$.

Example. *Given a set of n vertices, the family of all directed graphs G on V can be identified with a bit string $x \in \{0,1\}^{n \times n}$, where $x_{ij} = 1$ precisely when G has an edge between the ith vertex and the jth vertex. We can therefore consider the Hamming distance between two graphs, which give the number of edges upon which the graphs differ. If we want to consider how certain quantities on graphs change when we modify graphs by removing and adding some edges, such as the shortest path between points, or the minimum cut, then Fourier analysis gives a basic set of techniques to quantify this change, because the Hamming distance on the set of directed graphs with respect to this binary structure is exactly the number of edges upon which the graph's differ.*

Example. *The family of all subsets of a set with n elements can be identified with a Boolean string in $\{0,1\}^n$, which we can view as an indicator function which identifies which elements are present in the subset. The Hamming distance between two sets S_1 and S_2 is then precisely given by $\Delta(S_1, S_2) = \#(S_1 \Delta S_2)$, i.e. the cardinality of the symmetric difference of the two sets. Thus we see that Fourier analysis on Boolean functions might give applications to combinatorial problems in set theory.*

Example. *The values $\{0,1\}$ correspond to the truth values in the obvious way. Addition corresponds to the operation $\oplus$ of* exclusive or, *i.e. for two truth values P and Q, $P \oplus Q$ is true precisely when P is true, or Q is true, but* not both. *Fourier analytic methods can be used to identify logical properties via this correspondence.*

For notational convenience, the additive group structure on $\{0,1\}$ is isomorphic to the multiplicative group structure on $\{-1,+1\}$, where $0 \mapsto +1$, and $1 \mapsto -1$. This correspondence seems unintuitive at first, 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 the general convention of performing harmonic analysis over the multiplicative group $\mathbf{T}^n$ where possible. Rather than thinking of $\{0,1\}$ as the canonical group upon which we take the correspondence, we shall find that $\{-1,+1\}$ is notationally more important in Fourier analysis. For notational brevity, we let $\mathbf{B} = \{-1,+1\}$. Sometimes, $\mathbf{B}^n$ is called the *n dimensional Hamming cube*.

Determining the characters on the group $\mathbf{B}^n$ is easy, because the analysis reduces to the classical Fourier analysis on the Toral group $\mathbf{T}^n$. The characters of the toral group $\mathbf{T}^n$ are monomials $z \mapsto z_1^{m_1} \dots z_n^{m_n}$, where the m_k are integers. Because $\mathbf{B}^n$ is a closed subgroup of $\mathbf{T}^n$, the characters of $\mathbf{B}^n$ are restrictions of the characters on $\mathbf{T}^n$. However, characters on $\mathbf{T}^n$ can behave identically when restricted to $\mathbf{B}^n$. Indeed, the relation $x_i^2 = 1$, which holds for all $x_i \in \mathbf{B}$, implies that a monomial $x_1^{m_1} \dots x_n^{m_n}$ is trivial if the m_i are even numbers, so that the values of the restriction of a monomial to $\mathbf{B}^n$ only depend on the values of the integers $\{m_k\}$ modulo two. There are 2^n monomials with exponents modulo two, and since the harmonic analysis of abelian groups tells us there should be 2^n characters on $\mathbf{B}^n$, these monomials form distinct representatives of the class of all characters. We introduce a special notation for these characters:

- For $S \subset [n]$, the functions

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

  form distinct representatives of all $\mathbf{C}$-valued characters on $\mathbf{B}^n$. We will let

$$\sigma_S(x) = \sum_{i \in S} x_i$$

  denote the corresponding $\mathbf{F}_2$ valued character. These functions can be seen as measuring the *parity* of a particular subset sum of the coordinates of the input.

The basic theory of abelian harmonic analysis tells us that the characters x^S form an orthonormal basis for the Hilbert space $L^2(\mathbf{B}^n)$, so for any $f : \mathbf{B}^n \to \mathbf{R}$, we have a unique expansion

$$f(x) = \sum_S \widehat{f}(S) x^S \qquad g(x) = \sum_S \widehat{g}(S) \chi_S(x)$$

where $\widehat{f}(S)$ is the *Walsh-Fourier coefficients* corresponding to S. We shall ocasionally find it useful to analyze $\mathbf{B}^I$, where I is an arbitrary finite index set, in which case we extend our character notation to x^S and χ_S for $S \subset I$, and the Fourier expansion takes the same form

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

where S ranges over subsets of I.

An easy way to find Fourier coeffients, without any harmonic analysis, is to write a function in terms of more elementary functions with known Fourier coefficients. This is the binary equivalent of the technique for finding power series of holomorphic functions which can be expressed in terms of more basis power series. For instance, we can expand the any function $f : \mathbf{B}^n \to \mathbf{B}$ in its conjunctive normal form, calculating

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

This formula gives us a *particular* expansion of the function in monomials, and we know it therefore must be the *unique* expansion.

Example. *The maximum function* $\max : \mathbf{B}^2 \to \mathbf{B}$ *has an expansion*

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

This function is identified with the operation of conjucation $\{0,1\}^2 \to \{0,1\}$ *given by* $(x, y) \mapsto x \wedge y$. *To obtain this expansion, and a more general expansion for the maximum function on n variables note that we can write*

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

so that

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

Similarily, we consider the minimum function $\min : \mathbf{B}^n \to \mathbf{B}$ *on n bits, which corresponds to the disjunction on* $\{\top, \bot\}^n$. *Since* $\min(x) = -\max(-x)$,

$$\begin{aligned} \min(x) &= -\left((1 - 2^{1-n}) - 2^{1-n} \sum_{S \neq \varnothing} (-1)^{|S|} (-x)^S \right) \\ &= (2^{1-n} - 1) + 2^{1-n} \sum_{S \neq \varnothing} x^S \end{aligned}$$

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

For large n, the minimum and maximum functions are very close to constant functions, i.e. we have $\Delta(\max, 1) = \Delta(\min, -1) = 1$. Correspondingly, the Fourier coefficients of max and min corresponding to non-constant monomials are negligible. This reflects the general fact that if $\Delta(f, g)$ is small, then we should expect the Fourier coefficients of f and g to correspond to one another.

Example. *For $x, y \in \mathbf{B}$, $xy = 1$ precisely when $x = y$. This means that*

$$\mathbf{I}(x = y) = \frac{1 + xy}{2}$$

In general, we can express the indicator functions $\mathbf{I}(x = a) : \mathbf{B}^n \to \{0, 1\}$ via the expansion

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

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

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

Thus the degree *of this indicator function is equal by k, i.e. the only sets with nonzero Fourier coefficients have cardinality less than or equal to k.*

Example. *Let $f : \mathbf{B}^n \to \mathbf{B}$ be the discrete probability density function of a random vector whose components are independent $\{-1, 1\}$ Bernoulli distributions with means $\mu_1, \ldots, \mu_n \in [-1, 1]$. If X is a random variable with this distribution, then*

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

so $\mathbf{P}(X_i = 1) = 2^{-1}(\mu_i + 1) = P_i \in [0,1]$. *We then calculate*

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

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 dot product on* $\mathbf{F}_2^n$, *the function* $\mathbf{F}_2^n \times \mathbf{F}_2^n \to \mathbf{F}_2$ *given by*

$$x \cdot y = \sum_{i=1}^{n} x_i y_i$$

We can identify this dot product with a function $g : \mathbf{B}^n \times \mathbf{B}^n \to \mathbf{B}$. *The function* $(a,b) \mapsto ab$ *from* $\mathbf{F}_2 \times \mathbf{F}_2 \to \mathbf{F}_2$ *corresponds to the function* $(a,b) \mapsto \min(a,b) = 2^{-1}(a+b-ab-1)$, *so we find that*

$$g(x,y) = \frac{1}{2^n}\prod_{i=1}^{n}(x_i + y_i - x_i y_i - 1).$$

To find the Fourier expansion, we identify $\mathbf{B}^n \times \mathbf{B}^n$ *with* $\mathbf{B}^{\{1,2\}\times[n]}$ *in the natural way. For any subset* S *of indices on* $\{1,2\} \times [n]$ *can be decomposed as* $S = \{1\} \times S_1 \cup \{2\} \times S_2$, *where* $S_1, S_2 \subset [n]$. *Let* $\alpha(S)$ *be the cardinality of*

$S_1 \cap S_2$. *Then we have*

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

In particular, we find that

$$g(x,x) = \frac{1}{2^n}\sum_{S\subset\{1,2\}\times[n]}(-1)^{\alpha(S)}x^{S_1\triangle S_2}$$

The number of S_1 and S_2 with $S_1\triangle S_2 = [n]$ is exactly the number of bipartitions of $\{1,\ldots,n\}$. There are 2^n such bipartitions, and these partitions must be disjoint. This means the corresponding $\alpha(S)$ is always equal to 0, hence $x^{[n]}$ has a coefficient of 1. Conversely, if T is a proper subset of $[n]$, not containing some element x, then the set of S_1, S_2 can be divided in half into the family such that $x \in S_1$ and $x \in S_2$ and the family such that $x \notin S_1 \cup S_2$. The coefficient $(-1)^{\alpha(S)}$ on one family is the opposite of the coefficient of the other family, so when we sum over all possible coefficients, we find that the coefficient of $g(x,x)$ corresponding to x^T is zero. This makes sense, since $g(x,x) = x^{[n]}$.

Example. *The equality function EQ returns 1 only if all x_i are equal. Thus*

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

which makes sense since EQ is an even function, so the odd degree coefficients vanish. The not all equal function NAE returns 1 precisely when all x_i not all equal. We find

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

This function is very useful in analyzing the effectiveness of voting rules.

Example. *The sortedness function* $Sort : \mathbf{B}^4 \to \mathbf{B}$ *returns -1 if the bits in the input are monotonically increasing or monotonically decreasing. Since the function is even, all odd Fourier coefficients vanish. It will also help that Sort is invariant under the coordinate permutation* $(14)(23)$*, hence the Fourier coefficients obtained by swapping these coordinates are equal. To calculate the coefficients, it will be helpful to switch to calculating the coefficients of* $f = (1 - Sort)/2$*, which is now* $\{0,1\}$ *valued.* $f(x) = 1$ *if and only if* x *is one of the following 8 bit strings*

$$(-1,-1,-1,-1),(-1,-1,-1,+1),(-1,-1,+1,+1),(-1,+1,+1,+1)$$

$$(+1,+1,+1,+1),(+1,+1,+1,-1),(+1,+1,-1,-1),(+1,-1,-1,-1)$$

Hence the expected value of f *is* $1/2$*. We also calculate that*

$$\widehat{f}(\{1,2,3,4\}) = 0 \quad \widehat{f}(\{1,2\}) = \widehat{f}(\{3,4\}) = 1/4$$

$$\widehat{f}(\{1,3\}) = \widehat{f}(\{2,4\}) = 0 \quad \widehat{f}(\{1,4\}) = -1/4$$

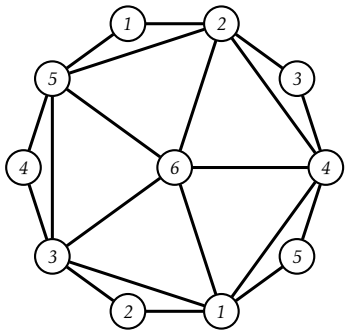
$$\widehat{f}(\{2,3\}) = 1/4$$

Thus $f(x) = (1/4)(2 + x_1x_2 + x_3x_4 + x_2x_3 - x_1x_4)$*, and so*

$$Sort(x) = 1 - 2f(x) = (1/2)(x_1x_4 - x_1x_2 - x_2x_3 - x_3x_4)$$

The x_1x_4 *term determines whether* x *should be the constant* $+1$ *or* -1 *term, and the rest offset the problem by pairwise comparisons.*

Example. *The hemi-icosohedron function* $HI : \mathbf{B}^6 \to \mathbf{B}$ *takes a set of labels for the vertices of the hemi-icosohedron graph below (embedable in the projective plane), and returns the number of faces labelled* $(+1,+1,+1)$*, minus the number of faces labelled* $(-1,-1,-1)$*, modulo 3*



The graph is homogenous, in the sense that there is a graph isomorphism mapping a vertex to any other vertex. In fact, one can obtain all isomorphisms by choosing which face one face will be mapped to, and specifying where two of the vertices on that face will go in the new face.

For any x, if we consider $-x$, then any face labelled $(+1,+1,+1)$ becomes a face labelled $(-1,-1,-1)$, and vice versa, so that $HI(-x) = -HI(x)$. Thus HI is an odd function, and all even Fourier coefficients vanish. Note that $HI(-1) = -1$, because all 10 faces are labelled $(-1,-1,-1)$, and $H(1) = 1$. If x has a single 1, then 5 of the faces are labelled $(-1,-1,-1)$, and none are labelled $(1,1,1)$, so $HI(x) = 1$. If x has exactly 2 ones, then we see that only two faces are labelled $(-1,-1,-1)$, and none are labelled $(+1,+1,+1)$, so $HI(x) = 1$. If x has three ones arranged in a triangle on the hemi-icosohedron, then the hemi-icosohedron has one face labelled $(+1,+1,+1)$, and none labelled $(-1,-1,-1)$, and hence $HI(x) = 1$. If x does not have three ones arranged in a triangle, then we find it must have three negative ones arranged in a triangle, hence $HI(x) = -1$. Since HI is odd, this calculates HI explicitly for all values of x.

Since there is an isomorphism mapping any vertex to any other vertex, the degree one Fourier coefficients of HI are all equal, and we might as well calculate the coefficient corresponding to x_1. Now

$$\widehat{HI}(1) = \mathbf{E}[HI(X)X_1] = \frac{\mathbf{E}[HI(X)|X_1 = 1] - \mathbf{E}[HI(X)|X_1 = -1]}{2}$$

Since HI is odd,

$$\mathbf{E}[HI(X)|X_1 = -1] = -\mathbf{E}[HI(-X)|X_1 = -1] = -\mathbf{E}[HI(X)|X_1 = 1]$$

hence $\widehat{HI}(1) = \mathbf{E}[HI(X)|X_1 = 1]$. If we let Y be the random variable denoting the number of $x_i = 1$, for $i \neq 1$, then

$$\begin{aligned}\mathbf{E}[HI(X)|X_1 = 1] &= (1/32)\sum_{k=0}^{5}\binom{5}{k}\mathbf{E}[HI(X)|X_1 = 1, Y = k] \\ &= (1/32)\left(\binom{5}{0} + \binom{5}{1} - \binom{5}{3} - \binom{5}{4} + \binom{5}{5}\right) \\ &= (1/32)(1 + 5 - 10 - 5 + 1) = -1/4\end{aligned}$$

since $\mathbf{E}[HI(X)|X_1 = 1, Y = 2] = (1/10)(5-5) = 0$. To simplify the calculation of coefficients of degree, note that we have graph isomorphisms mapping

any triangle to any other triangle, hence the Fourier coefficients of degree 3 corresponding to a triangle are all equal. If Y is the number of $x_i = 1$, for $i \neq 1, 2, 3$, then by applying the same technique as when we were calculating $\mathbf{E}[HI(X)|X_1 = 1]$, we find

$$\begin{aligned}
\mathbf{E}[HI(X)|X_1 = +1, X_2 = +1, X_3 = +1] &= (1/8)(1 - 3 - 3 + 1) = -1/2 \\
\mathbf{E}[HI(X)|X_1 = +1, X_2 = +1, X_3 = -1] &= (1/8)(1 + 1 - 2 - 3 - 1) = -1/2 \\
\mathbf{E}[HI(X)|X_1 = +1, X_2 = -1, X_3 = -1] &= -\mathbf{E}[HI(X)|X_1 = +1, X_2 = +1, X_3 = -1] \\
&= 1/2 \\
\mathbf{E}[HI(X)|X_1 = -1, X_2 = -1, X_3 = -1] &= -\mathbf{E}[HI(X)|X_1 = +1, X_2 = +1, X_3 = +1] \\
&= 1/2
\end{aligned}$$

We can then interpolate these results to find

$$\widehat{f}(\{1, 2, 3\}) = (-1/2 + 3/2 + 3/2 - 1/2)(1/8) = 1/4$$

Now the sum of the squares of the Fourier coefficients we have calculated are $10(1/4)^2 + 6(1/4)^2 = 1$, so that all other Fourier coefficients are zero. Thus

$$f(x) = 1/4(e_\Delta - e_1)$$

When e_1 is the first symmetric polynomial (the sum of all monomials of degree one), and e_Δ is the sum of all degree three monomials which correspond to triples of vertices in the hemiicosohedron which form triangles.

1.2 Alternate Expansions on $\mathbf{F}_2^n$

There are certainly other bases for representing real-valued functions on the domain $\mathbf{F}_2^n$, as alternatives to the Fourier expansion. For $f : \mathbf{F}_2^n \to \mathbf{R}$, we also have an expansion $f(x) = \sum a_S x^S$, where $x^S = \prod_{i \in S} x_i$ is equal to 1 when all x_i are equal to one, for $i \in S$, and is equal to 0 otherwise. This is very different than the measures of parity on $\mathbf{B}^n$. These monomials are certainly independent (a general property of polynomials over a field, in this case elements of $\mathbf{F}_2[x_1, \ldots, x_n]$), and span a subspace of functions of dimension 2^n. It follows that all real-valued functions on $\mathbf{F}_2^n$ have a unique expansion as monomials of this form, because these functions form a space of dimension 2^n.

We can construct such an expansion by induction, noting that for functions $f : \mathbf{F}_2 \to \mathbf{R}$, $f(x) = f(0) + x[f(1) - f(0)]$, and if $f : \mathbf{F}_2^{n+1} \to \mathbf{R}$, then by induction there are $a_S, b_S \in \mathbf{R}$ such that

$$f(x,0) = \sum a_S x^S \qquad f(x,1) = \sum b_S x^S$$

and then

$$f(x,y) = f(x,0) + y[f(x,1) - f(x,0)] = \sum a_S x^S + \sum (b_S - a_S) y x^S$$

This expansion of functions on $\mathbf{F}_2^n$ are an interesting phenomenon, but aren't too useful to the study of harmonic analysis. Firstly, the x^S do not form an orthonormal basis for the class of real-valued functions, which makes them less useful. They do diagonalize the multiplication operators $(\sigma_y f)(x_1, \ldots, x_n) = f(x_1 y_1, \ldots, x_n y_n)$, for $y \in \mathbf{F}_2^n$, but it turns out that these operators do not really occur in many applications; these operators just annihilate certain coordinates of f, and these are very simple to understand without Fourier analysis. However, in certain cases this expansion is useful for obtain the Fourier expansion of some function. Because we have an expansion

$$x^S = (1/2)^{|S|} \sum_{T \subset S} (-1)^{|T|} \chi_T(x)$$

If $f : \mathbf{F}_2^n \to \mathbf{R}$ is expanded as a polynomial $f(x) = \sum_S a_S x^S$, and is also expanded as a sum of characters $f(x) = \sum_S b_S \chi_S(x)$, then we find that

$$b_T = (-1)^{|T|} \sum_{T \subset S} \frac{a_S}{2^{|S|}}$$

In particular, we find that the degrees of both expansions are the same, because if T is a maximal set with $a_T \neq 0$, then $b_T = (-1/2)^{|T|} a_T \neq 0$, and for any $T \subsetneq U$, $b_U = 0$.

1.3 Fourier Analysis and Probability Theory

Many interesting results of Boolean Fourier analysis occur in the context of statistical problems over discrete domains. Thus it is of interest to interpret the main tools and results of probability theory in the context of

Boolean analysis. Unless otherwise stated, we will assume a random variable X taking values over $\mathbf{B}^n$ is uniformly distributed over the domain. The probability theory is often just normalized combinatorics, but it is still a useful source of intuition and elegant notation, as well as a source for problems to be solved using Boolean harmonic analysis, so it is useful to mention it.

The canonical way to introduce probability to the study of functions over some space is to introduce a probability measure over the space, and then to look at the integration theory which gives a metric on the functions on the space. In this case, we can define an inner product on the space of real-valued Boolean functions by letting

$$\langle f, g \rangle = \frac{1}{2^n} \sum f(x)g(x) = \mathbf{E}[f(X)g(X)]$$

The characters on $\mathbf{B}^n$ form an orthonormal basis to this inner product, and it therefore follows that for any Boolean function f,

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

The Fourier coefficients of f correspond to the average value of f relative to the parity of S. For instance, a Boolean-valued function $f : \mathbf{B}^n \to \mathbf{B}$ is *unbiased*, that is, it takes values +1 and -1 with equal probability, if and only if $\widehat{f}(\varnothing) = 0$. The standard Hilbert space results give rise to probabilistic interpretations. Parseval's identity is just the probabilistic equation $\mathbf{E}[f(X)^2] = \sum \widehat{f}(S)^2$, which has a simple corollary that

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

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

Hence expectations, variances, and covariances are directly related to the Fourier coefficients of the Boolean function in question, and we can often answer questions about these quantities by relating them to the Fourier coefficients.

For a *Boolean-valued* map $f : \mathbf{B}^n \to \mathbf{B}$, we find that $f^2 = 1$, so $\|f\|_2^2 = 1$, and therefore

$$\sum \widehat{f}(S)^2 = 1$$

It therefore follows that the squares of the Fourier coefficients generate a probability distribution over subsets of $[n]$. We call this distribution the **spectral sample** of f. We define the weight of a function $f : \mathbf{B}^n \to \mathbf{R}$ of degree k to be

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

If f is Boolean-valued, then $\mathbf{W}^k(f) = \mathbf{P}(|S| = k)$ where S is a random set drawn from the spectral sample induced by f. $\mathbf{W}^k(f)$ is also the L^2 norm of the function

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

which can be seen as a certain truncation of f over the degree k terms.

If f and g are both Boolean-valued maps, then we can write

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

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

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

Thus the variance of Boolean valued functions is very closely related to the degree to which a function differs from a constant function in the Hamming distance. We therefore have the following result.

Lemma 1.1. *If $f : \mathbf{B}^n \to \mathbf{B}$ is a Boolean function, and $\varepsilon = \min[d(f,1), d(f,-1)]$, 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 divide by $4\min(x,1-x)$ and restate the inequality as $1/2 \leqslant \max(x,1-x) \leqslant 1$, which is obvious. The lower bound is tight for $\varepsilon \sim 1/2$, where the variance is maximal, and the upper bound is tight for extreme values of ε, where the variance is negligible. □

In general domains we intuitively view $\mathbf{V}(f)$ as a measure of how constant the function f is, and the result above gives a precise instance of this heuristic for Boolean functions. In the context of the Fourier expansion of the variance of a function this makes sense, we obtain that if $\varepsilon = \min[d(f,1), d(f,-1)]$, then $\widehat{f}(\varnothing)^2 = 1 - 4\varepsilon + 4\varepsilon^2 = 1 - 4\varepsilon + O^+(\varepsilon^2)$, so a function is virtually constant if and only if a majority of its Fourier weight is concentrated on the degree zero coefficient.

Example. *Consider a function f chosen uniformly randomly from the set of all Boolean functions on $\mathbf{B}^n$. For any function $g : \mathbf{B}^n \to \mathbf{B}$, 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 $\widehat{f}(S)$ is now a real-valued random variable. Now because we have a bijection amongst the set of all Boolean functions, mapping g to $-g$, we find

$$\mathbf{E}[\widehat{f}(S)] = \frac{1}{2^{2^n}} \sum_g \widehat{g}(S) = \frac{1}{2^{2^n}} \sum_g \widehat{(-g)}(S) = -\frac{1}{2^{2^n}} \sum \widehat{g}(S) = -\mathbf{E}[\widehat{f}(S)]$$

Hence $\mathbf{E}[\widehat{f}(S)] = 0$. What's more, given that $X \neq Y$, the probability that $f(X) = f(Y)$ is independent of X and Y, hence

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

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 $\mathbf{V}_f[\widehat{f}(S)] = 1/2^n$. Thus functions on average have no Fourier coefficient on any particular set S, and most functions have very small Fourier coefficients.

1.4 Testing Linearity

A function $f : \mathbf{B}^n \to \mathbf{B}$ is linear if $f(xy) = f(x)f(y)$ for all $x, y \in \mathbf{B}^n$, 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. This is guaranteed, because the minimal description length of a general function f requires a description length of $\Theta(2^n)$ bits. If an algorithm determining linearity does not check the entire description of the function f, we can modify f to be non-linear function on inputs that the linearity tester doesn't look at, and the algorithm will not be able to distinguish between these two inputs. The problem of testing linearity is a scenario in a family of problems 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 will address the more general theory in a later chapter.

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 a 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 if a function is likely to succeed, then the function is guaranteed to be close to a linear function.

The simplest version of linearity testing runs using one random query as a test – it just takes a pair of uniformly random inputs, and tests the linearity of this function against these inputs. This is known as the Blum-Luby-Rosenfeld algorithm, or BLR test for short. It turns out that the sucess of this method is directly related to how similar a function is to a linear character. There are two candidates to define what it means to be 'approximately linear'. For a subset A of Boolean-valued functions, we shall let $d(f, A) = \inf_{g \in A} d(f, g)$ denote the minimum distance from f to A, and we will let $\mathcal{C}$ denote the subset of Boolean functions which are characters, i.e. of the form x^S for some $S \subset [n]$.

So how do we quantify the ability for the BLR test to succeed. To begin with, there are two ways we could say a function f is 'close' to being linear.

- $f(xy) = f(x)f(y)$ for a large majority of inputs x and y. Essentially, this means exactly that $f(XY) = f(X)f(Y)$ holds with high probability, so this property is equivalent to the BLR test being accepted with high probability.
- There is a linear functional g such that $f(x) = g(x)$ for a large number of inputs x. This means exactly that $d(f, (\mathbf{F}_2^n)^*)$ is small, so f is close to some character.

The two notions are certainly related, but a direct relation seems difficult to quantify. One way is easy to bound. If g is linear, and $d(f,g) = \varepsilon$, then a standard union bound inequality guarantees that

$$\begin{aligned}\mathbf{P}[f(X)f(Y) = f(XY)] &\geqslant \mathbf{P}[f(X) = g(X), f(Y) = g(Y), f(XY) = g(XY)] \\ &\geqslant 1 - 3\varepsilon\end{aligned}$$

But just because $f(xy) = f(x)f(y)$ holds for a large number of x, y doesn't seem to imply that f is close to any *particular* linear function. The only thing which gives us hope is that the set of characters is very discrete with respect to the Hamming distance. If ϕ and ψ are two distinct characters, then $d(\phi, \psi) = 1/2$, so it is difficult to agree with both functions at once. What's more, since $\widehat{f}(S) = \mathbf{E}[f(X)X^S] = 1 - 2d(f, x^S)$, the identity $\sum \widehat{f}(S)^2 = 1$ shows that $d(f, x^S)$ isn't large for too many distinct sets S.

Theorem 1.2. *For any function $f : \mathbf{B}^n \to \mathbf{B}$,*

$$\mathbf{P}(\textit{BLR algorithm rejects } f) \geqslant d(f, \mathcal{C})$$

So that a function is accepted with probability ε if and only if $d(f, \mathcal{C}) \leqslant 1 - \varepsilon$, so we can select a single linear character $g : \mathbf{B}^n \to \mathbf{B}$ with $d(f,g) \leqslant 1 - \varepsilon$.

Proof. Note that

$$\mathbf{I}[f(XY) \neq f(X)f(Y)] = \frac{1 - f(X)f(Y)f(XY)}{2}$$

Hence

$$\begin{aligned}
\mathbf{P}[\text{BLR rejects } f] &= \mathbf{E}\left[\frac{1 - f(X)f(Y)f(XY)}{2}\right] \\
&= \frac{1}{2} - \frac{1}{2}\mathbf{E}_X[f(X)\mathbf{E}_Y[f(Y)f(XY)]] \\
&= \frac{1}{2} - \frac{1}{2}\mathbf{E}_X[f(X)(f * f)(X)] \\
&= \frac{1}{2} - \frac{1}{2}\sum \widehat{f}(S)^3 \\
&\geqslant \frac{1}{2} - \frac{1}{2}\max \widehat{f}(T)\sum \widehat{f}(S)^2 \\
&= \frac{1}{2} - \frac{1}{2}\max \widehat{f}(T)
\end{aligned}$$

For each S, $\widehat{f}(S) = 1 - 2d(f, x^S)$, so if $d(f, x^S) > \varepsilon$ for all S, we find that $\widehat{f}(S) < 1 - 2\varepsilon$, hence

$$\mathbf{P}[\text{BLR rejects } f] > \frac{1}{2} - \frac{1 - 2\varepsilon}{2} = \varepsilon$$

and this completes the proof. □

Note that the BLR 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, $f(x) = x^S$, but we do not know for which x this occurs. The problem is that x is a fixed quantity, rather than varying over many values of the function, and therefore our test is easy to break. Note, however, that $y^S = x^S(xy)^S$, and if we let X be a random quantity, then $f(Xy)$ and $f(y)$ will likely be equal to $(Xy)^S$ and X^S, with a probability that is feasible to determine.

Theorem 1.3. *If $d(f, x^S) \geqslant \varepsilon$, then for any x,*

$$\mathbf{P}[f(xY)f(Y) = x^S] \geqslant 1 - 2\varepsilon$$

Proof. We just apply a union bound to conclude

$$\mathbf{P}(f(xY)f(Y) = x^S) \geqslant \mathbf{P}(f(xY) = x^S Y^S, f(Y) = Y^S) \geqslant 1 - 2\varepsilon$$

and in this circumstance, we find $f(yX)f(X) = y^S$. □

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 the theory of property testing and complexity theory, which we will discuss later.

The inequality in our analysis of the BLR test is tight, provided that all Fourier coefficients are reasonably equal to one another. Analyzing the gap in the inequality, we find

$$\begin{aligned}\max \widehat{f}(T) \sum \widehat{f}(S)^2 - \sum \widehat{f}(S)^3 &= \sum \widehat{f}(S)^2 [\max \widehat{f}(T) - \widehat{f}(S)] \\ &\leqslant \max \widehat{f}(T) - \min \widehat{f}(S)\end{aligned}$$

If all Fourier coefficients are equal, then the inequality is an equality. Though this isn't strictly the spectra of a *Boolean-valued* function (it's the spectra of the indicator function $\mathbf{I}(x = 1)$), we can find Boolean functions f whose Fourier coefficients which have as small a variation as desired. For instance, the difference between the maximum and minimum coefficients of the inner product functions $f(x, y) = (-1)^{\langle x,y \rangle}$ on $\mathbf{F}_2^n$ is 2^{1-n}. Thus we have found the best dimension independant constant that holds over all functions f. However, if one of the $\widehat{f}(S)$ is much larger than the others, so f is very close to x^S, then we can obtain a better bound, and thus a much tighter analysis of the BLR test.

Theorem 1.4. *If $d(f, x^S) = \varepsilon$, then the BLR test rejects f with probability at least $3\varepsilon - 10\varepsilon^2 + 8\varepsilon^3$.*

Proof. Note that

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

and also for $T \neq S$,

$$\widehat{f}(T) = 1 - 2d(f, x^T) \leqslant 1 - 2(d(f, x^S) - d(x^T, x^S)) = 2\varepsilon$$

Hence

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

The bound here is much tighter than the other calculations. Indeed the difference in the inequality is

$$\sum_{T\neq S}\varepsilon\widehat{f}(T)^2 - (1/2)\widehat{f}(T)^3 = \sum_{T\neq S}\widehat{f}(T)^2[\varepsilon - \widehat{f}(T)/2] \leqslant \max(\varepsilon - \widehat{f}(T)^2/2) \leqslant \varepsilon$$

So for small ε the bound is tight. □

The converse of this statement is that if the BLR test accepts f with probability greater than $1 - 3\varepsilon + O(\varepsilon^2)$, then f is ε close to being linear. Thus for small ε the bound on similarity to a linear function is tighter than we initially calculated. This makes sense because, when f is very close to a linear function, it should be difficult for $f(xy) = f(x)f(y)$ *not* to hold.

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 $g_n(x) = \mathbf{I}(x = 1)$ (which we saw made the inequality go tight in the first analysis). 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 any $0 < \delta < C$, 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 as $n \to \infty$.

1.5 The Walsh-Fourier Transform

Historically, the study of the Fourier transform on $\mathbf{F}_2^n$ emerged from a novel analysis of the Fourier expansions of $L^2[0,1]$. In this section we consider this development to see how the theory historically developed. Consider the product group $\mathbf{F}_2^\infty = \prod \mathbf{F}_2$, which is a compact abelian group. We have the standard embeddings $i_n : \mathbf{F}_2^n \to \mathbf{F}_2^\infty$ and projections $\pi_n : \mathbf{F}_2^\infty \to \mathbf{F}_2^n$, defined by

$$i_n(x_1,\dots,x_n) = (x_1,\dots,x_n,0,0,\dots) \qquad \pi_n(x_1,x_2,\dots) = (x_1,\dots,x_n)$$

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 $S = \lim S_n$ is not a bounded subset of the integers, then the limit above can be allowed to oscillate infinitely between -1 and 1, hence f cannot be a *continuous* character on $\mathbf{F}_2^\infty$. We therefore find that characters on $\mathbf{F}_2^\infty$ correspond to finite subsets of positive integers, and topologically the character group $(\mathbf{F}_2^\infty)^*$ is 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 $[0,1]$ can be viewed as defining a measure preserving map between $\mathbf{F}_2^\infty$ and $[0,1]$, where an infinite sequence $(x_1, x_2, \dots)$ maps to $\sum_{m=1}^\infty x_m/2^m$. This map is injective almost everywhere, and therefore defines a measure preserving isomorphism, and is certainly surjective. Thus we can find a uniform distribution over $[0,1]$ by taking an infinite binary expansion, where the coefficients are obtained by flipping a fair coin infinitely many times.

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. These functions are known as **Walsh function**, and every square-summable function has a unique Walsh expansion

$$f(x) = \sum_{n=0}^{\infty} a_n w_n(x) = \sum_{\substack{S \subset \mathbf{N} \\ |S| < \infty}} \widehat{f}(S) w^S(x)$$

This is essentially where the beginnings of the study of Boolean expansions began, back in 1920. The Walsh expansions have very good L^p truncation properties, 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 the purposes of discrete Boolean analysis, 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 $\{X_k\}$ of functions from $\mathbf{F}_2^n \to \mathbf{R}$ defined by $X_k = \mathbf{I}(x_1 = k_1, \dots, x_n = k_n)$, and a basis $\{Y_k\}$ of functions from $\mathbf{F}_2^*$ to $\mathbf{R}$ defined by $Y_k = \chi_{\{i : k_i = 1\}}$. The Walsh transform $\mathcal{W}$ is effectively a map from $\mathbf{R}^{\mathbf{F}_2^n}$ to $\mathbf{R}^{(\mathbf{F}_2^n)^*}$, these bases induce a matrix representation W of the transform. We have

$$\mathcal{W}(X_l) = \frac{1}{2^n} \sum_S (-1)^{\sum_{m \in S} l_m} \chi_S = \frac{1}{2^n} \sum_k (-1)^{\sum l_i k_i} Y_k$$

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 $W = H_n/2^n$, because, as can be proved by induction, $H^n_{ab} = (-1)^{\sum a_i b_i}$. 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 maximum number of additions and subtractions required to calculate the Walsh transform a n-dimensional Boolean function, then $t_{n+1} \leqslant 2t_n + 2n$, with $t_1 = 0$, hence

$$t_n \leqslant 2^{n+1} \sum_{k=2}^{n} k \left(\frac{1}{2}\right)^k \leqslant n2^{n+1} \sum_{k=2}^{n} \left(\frac{1}{2}\right)^k \leqslant n2^n$$

so the calculation is possible in $n2^n$ operations, 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 requires $m \lg m$ operations. This is essentially the fastest way to compute the Fourier coefficients of an arbitrary boolean function.

Chapter 2

Social Choice Functions

We now interpret boolean functions in a particular non-abstract scenario in the theory of social choice. Here a Boolean $f : \mathbf{B}^n \to \mathbf{B}$ is employed as a device to decide the result of a two-candidate election. A group of n people choose between two candidates, labelled -1 and $+1$, which are fed into the function f as an element x of $\mathbf{B}^n$, and $f(x)$ determines the overall outcome of the vote. In this situation it is natural to find voting rules with various properties relavent to promoting a fair election, and these properties also have an interest in the general context of the harmonic analysis of Boolean functions.

Example. *The standard Boolean function used as a voting rule is the majority function* $Maj_n : \mathbf{B}^n \to \mathbf{B}$, *which outputs the candidate who got the majority of votes. This is always well defined with an odd number of voters, but there are issues with ties when an even number of voters is given. A function is called a majority function if it always chooses the candidate with the majority of votes, on inputs where this property is well defined.*

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

Example. *The dictator rule* $\chi_i : \mathbf{B}^n \to \mathbf{B}$ *defined by* $\chi_i(x) = x_i$ *places the power of a binary decision in the hands of a single person.*

Example. *All of these voting rules can be quantified as a version of the weight majority, or linear threshold function, those functions* f *which can be defined,*

for some $a_0, \ldots, a_n \in \mathbf{R}$, by the equation

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

Thus each voter has a certain voting power, specified by the a_i, and there is an additional bias a_0 which pushes the vote in a certain candidate's favour.

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*

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

where each $x_i \in \mathbf{B}^{n^{d-1}}$. 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*

$$Tribes_{ws} : \mathbf{B}^{sw} \to \mathbf{B}$$

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

where each $x_i \in \mathbf{B}^w$. The number of ways we can fail to pass each individual subelection is $2^w - 1$, so that $\mathbf{P}(Tribes_{ws} = 1) = (2^w - 1)^s / 2^{sw} = (1 - 2^{-w})^s$, and so $\mathbf{E}[Tribes_{ws}] = 2(1 - 2^{-w})^s - 1$. Indeed, fixing w and setting $(1 - 2^{-w})^s = 1/2$, we find that the optimal values of s which causes the voting rule to be unbiased satisfy

$$1/s = -\frac{\ln(1 - 2^{-w})}{\ln 2} = \frac{2^{-w} + O^+(2^{-2w})}{\ln 2}$$

Hence

$$s = \frac{\ln 2}{2^{-w} + O^+(2^{-2w})} = \frac{2^w \ln 2}{1 + O^+(2^{-w})} = 2^w \ln 2 - O(1)$$

If we let $s = 2^w \ln 2$ exactly (which of course, isn't really possible, but as $w \to \infty$ we can always choose integer values of s asymptotically equivalent to $2^w \ln 2$), then the average value of $Tribes_{w2^w}$, is obtained by noting that if $a = \ln 2$, then

$$(1 - 1/x)^{ax} = 1/2 + O(1/x)$$

Hence $\mathbf{E}[Tribes_{w(\ln(2)2^w)}] = O(2^{-w})$. *If we let* $n = w\ln(2)2^w$, *then the expected value is* $O(\log n/n)$. *To calculate the variance, we use the formula for Boolean functions to determine*

$$\begin{aligned}\mathbf{V}[Tribes_{ws}] &= 4\mathbf{P}(Tribes_{ws}(X) = 1)\mathbf{P}(Tribes_{ws}(X) = -1) \\ &= 4(1-2^{-w})^s[1-(1-2^{-w})^s]\end{aligned}$$

The variance for $s = w\ln(2)$ *is therefore* $4[1/2 + O(2^{-w})]^2 = 1 + O(2^{-w})$.

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 $\mathbf{B}^n$ 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 $f : \mathbf{B}^n \to \mathbf{B}$ is a function which is monotone and symmetric. Then $f(x) = g(\#\{i : x^i = 1\})$ for some monotone function $g : \{0,\dots,n\} \to \mathbf{B}$. 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 choice of N is impossible. □

2.1 Influence

Given a voting rule $f : \mathbf{B}^n \to \mathbf{B}$, 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, given $x \in \mathbf{B}^n$, we define $x^{\oplus i} \in \mathbf{B}^n$ 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. The uniform distribution also gives us values which work as normalized combinatorial quantities. 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. In particular, this implies that

$$\begin{aligned}\mathbf{P}(f(X^{\oplus i}) \neq f(X)) &= \mathbf{P}(f(X^{\oplus i}) = 1, f(X) = -1) + \mathbf{P}(f(X^{\oplus i}) = -1, f(X) = 1) \\ &= 2\mathbf{P}(f(X^{\oplus i}) = 1, f(X) = -1)\end{aligned}$$

so when calculating the influence of a function, it suffices to identify the cases where $f(X) = -1$ and $f(X^{\oplus i}) = 1$.

Example. *The dictator function χ_i satisfies $Inf_i(\chi_j) = \delta_{ij}$. 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 And_n only at the points 1 and $1^{\oplus i}$, so that $Inf_i(And_n) = 2/2^n = 2^{1-n}$.*

Example. *If $f(x) = sgn(a_0 + \sum a_j x_j)$ with $a_j \geqslant 0$, then whenever $X_i = -1$, then $f(X^{\oplus i}) \geqslant f(X)$ so the only circumstances where $f(X) = -1$ and $f(X^{\oplus i}) = 1$ is where $X_i = -1$. In particular, this can only occur if the values of X_j satisfy the inequality $-a_i < a_0 + \sum_{j \neq i} a_j X_j < a_i$. As an example, consider the voting rule*

$$f(x) = sgn(-58 + 31x_1 + 31x_2 + 28x_3 + 21x_4 + 2x_5 + 2x_6)$$

Since the coefficient of x_1 is equal to the coefficient of x_2, $Inf_1(f) = Inf_2(f)$. We find

$$\begin{aligned} Inf_1(f) = Inf_2(f) &= \mathbf{P}(-31 < -58 + 31x_2 + 28x_3 + 21x_4 + 2x_5 + 2x_6 < 31) \\ &= \mathbf{P}(27 < 31x_2 + 28x_3 + 21x_4 + 2x_5 + 2x_6 < 89) \\ &= (1/2)\mathbf{P}(-4 < 28x_3 + 21x_4 + 2x_5 + 2x_6 < 58) \\ &\quad + (1/2)\mathbf{P}(58 < 28x_3 + 21x_4 + 2x_5 + 2x_6 < 120) \\ &= (1/4)\mathbf{P}(-32 < 21x_4 + 2x_5 + 2x_6 < 30) \\ &\quad + (1/4)\mathbf{P}(24 < 21x_4 + 2x_5 + 2x_6 < 86) \\ &= (1/4) + (1/4)(1/8) = 9/32 \end{aligned}$$

We also find

$$\begin{aligned} Inf_3(f) &= \mathbf{P}(-28 < -58 + 31x_1 + 31x_2 + 21x_4 + 2x_5 + 2x_6 < 28) \\ &= \mathbf{P}(30 < 31x_1 + 31x_2 + 21x_4 + 2x_5 + 2x_6 < 86) \\ &= (1/4)\mathbf{P}(-32 < 21x_4 + 2x_5 + 2x_6 < 24) = (1/4)(7/8) = 7/32 \end{aligned}$$

$$\begin{aligned} Inf_4(f) &= \mathbf{P}(37 < 31x_1 + 31x_2 + 28x_3 + 2x_5 + 2x_6 < 79) \\ &= (1/4)\mathbf{P}(-25 < 28x_3 + 2x_5 + 2x_6 < 17) \\ &= (1/8)\mathbf{P}(3 < 2x_5 + 2x_6 < 45) \\ &= 1/32 \end{aligned}$$

Finally, we calculate the influence of x_5 and x_6.

$$\begin{aligned} Inf_5(f) = Inf_6(f) &= \mathbf{P}(56 < 31x_1 + 31x_2 + 28x_3 + 21x_4 + 2x_5 < 60) \\ &= (1/4)\mathbf{P}(-6 < 28x_3 + 21x_4 + 2x_5 < -2) = 1/32 \end{aligned}$$

Hence the function has total influence 28/32.

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

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

As n increases, the influence of each coordinate decreases monotonically, because

$$\begin{aligned}(2/2^{n+2})\binom{n+1}{\frac{n+1}{2}} &= \frac{1}{2^{n+1}}\left(\binom{n}{\frac{n+1}{2}} + \binom{n}{\frac{n-1}{2}}\right) \\ &= \frac{1}{2^{n+1}}\left(\binom{n-1}{\frac{n+1}{2}} + \binom{n-1}{\frac{n-1}{2}} + \binom{n-1}{\frac{n-1}{2}} + \binom{n-1}{\frac{n-3}{2}}\right) \\ &\leqslant \frac{1}{2^{n+1}}\left(4\binom{n-1}{\frac{n-1}{2}}\right) = \frac{1}{2^{n-1}}\binom{n-1}{\frac{n-1}{2}}\end{aligned}$$

Applying Stirling's approximation $m! = \sqrt{2\pi m}(m/e)^m(1 + O^+(1/m))$, we find

$$\begin{aligned}Inf_i(Maj_n) &= \frac{1}{2^{n-1}}\binom{n-1}{\frac{n-1}{2}} \\ &= \frac{(n-1)!}{2^{n-1}\left[\left(\frac{n-1}{2}\right)!\right]^2} \\ &= \frac{\sqrt{2\pi(n-1)}\left(\frac{n-1}{e}\right)^{n-1}(1 + O^+(1/n))}{2^{n-1}\pi(n-1)\left(\frac{n-1}{2e}\right)^{n-1}(1 + O^+(1/n))^2} \\ &= \sqrt{\frac{2}{\pi(n-1)}}\frac{1 + O^+(1/n)}{1 + O^+(1/n)} \\ &= \sqrt{\frac{2}{\pi n}}\sqrt{\frac{n}{n-1}}(1 + O^+(1/n))\end{aligned}$$

and

$$\begin{aligned}n\left(\sqrt{\frac{n}{n-1}}(1 + O^+(1/n)) - 1\right) &= \left(\sqrt{\frac{n}{n-1}} - 1\right)n + O^+(1) \\ &= \frac{n}{\sqrt{n-1}(\sqrt{n} + \sqrt{n-1})} + O^+(1) \\ &\leqslant \frac{1}{2} + O^+(1)\end{aligned}$$

Hence $\sqrt{\frac{n}{n-1}}(1+O^+(1/n)) = 1+O^+(n^{-1})$*, and so*

$$Inf_i(Maj_n) = \sqrt{\frac{2}{\pi n}}\sqrt{\frac{n}{n-1}}(1+O^+(1/n)) = \sqrt{\frac{2}{\pi n}} + O^+(n^{-3/2})$$

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

Example. *To calculate the influence of the tribes function, we note that conditioning on* $X_i = 1$*, the set of events where* $Tribes_{ws}(X) \neq Tribes_{ws}(X^{\oplus i})$ *is equal to the set of events where all the other tribes reject the vote, and the tribe containing* i *all* -1 *except for* X_i *itself. The number of possible choices here is therefore* $(2^w-1)^{s-1}$*, because the tribe containing* i *is essentially fixed, and the probability that* i *changes the vote gives us the influence as*

$$Inf_i(Tribes_{ws}) = \frac{(2^w-1)^{s-1}}{2^{ws-1}} = \frac{(1-2^{-w})^{s-1}}{2^{s-1}}$$

If $s = \ln(2)2^w$*, then*

$$Inf_i(Tribes_{ws}) = \frac{(1-2^{-w})^s}{2^{s-1}(1-2^{-w})} = \frac{1/2+O(2^{-w})}{2^{s-1}(1-2^{-w})} = O(2^{-s})$$

If $n = sw$*, then* $O(2^{-s}) = O(2^{-n/\log(n)})$*.*

To connect the influence of a Boolean function to it's Fourier expansion, we must come up with an analytic expression for the influence. To do this, we find an operator strongly related to the quantity of influence, and which is diagonalized by the Fourier monomials. 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, $D_i f$ 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

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

Since D_i is defined on all real-valued functions, we can use these formulas to extend the definition of influence for all *real-valued* Boolean functions. 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 $\mathrm{Inf}_i(f)$, and this makes the influence a much more easy quantity to quantify. In particular, we can now already see that an index i is nonessential if and only if $\widehat{f}(S) = 0$ for all sets S such that $i \in S$. Furthermore, we obtain the powerful inequality that $\mathrm{Inf}_i(f) \leqslant \mathbf{V}(f)$.

We shall dwell on the Laplacian for slightly longer than the other operators, since we will find it can be generalized to other domains in probability theory. 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 either always increasing in the ith direction, or always decreasing. This means exactly that $D_i f \geqslant 0$, or $D_i f \leqslant 0$, and in this case the bound above is actually an equality, so that $\mathrm{Inf}_i(f) = \widehat{f}(i)$. 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. In particular, the Majority function maximizes this influence among the monotone, transitive symmetric voting rules (up to a constant offset).

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

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

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 required inequality. □

It might be possible to make this inequality tighter. The inequality is only tight when the majority of the Fourier weight of f is concentrated on the degree one coefficients. But if $\mathbf{W}^1(f) \geqslant 1 - \varepsilon$, then the Friedgut Kalai Naor theorem (to be proved later) implies that f is $O(\varepsilon)$ close to $\pm x^i$ for some index i, a dictator or its negation, in which case for small ε it is unlikely that f can be both monotone and transitive symmetric. Thus the inequality is always slightly loose. However, I can't think of a way to tighten this inequality at this time.

Though most Boolean functions are not monotone, there is a way to obtain a monotone function closely related to any Boolean function $f : \mathbf{B}^n \to \mathbf{B}$. The **i'th polarization** of f is the Boolean function f^{σ_i} defined by

$$f^{\sigma_i}(x) = \begin{cases} \min(f(X), f(X^{\oplus i})) & X_i = -1 \\ \max(f(X), f(X^{\oplus i})) & X_i = 1 \end{cases}$$

Then f^{σ_i} is monotone in the ith direction, and if f is monotone in the jth direction, then so too is f^{σ_i}, because if $X_i = -1$, then

$$\begin{aligned} f^{\sigma_i}(X^{j\to -1}) &= \min(f(X^{j\to -1}), f(X^{j\to -1, i\to 1})) \\ &\leqslant \min(f(X^{j\to 1}), f(X^{j\to 1, i\to 1})) = f^{\sigma_i}(X^{j\to 1}) \end{aligned}$$

Thus $f^{\sigma_1\sigma_2\cdots\sigma_n}$ is a monotone Boolean function in every variable. Because we are essentially only rearranging elements of the domain, we find that $\mathbf{E}[f^{\sigma_i}(X)] = \mathbf{E}[f(X)]$ and $\|f^{\sigma_i}\|_p = \|f\|_p$. We also have $\mathrm{Inf}_i(f^{\sigma_i}) = \mathrm{Inf}_i(f)$ because

$$\mathbf{P}(f^{\sigma_i}(X) \neq f^{\sigma_i}(X^{\oplus i})) = \mathbf{P}(f(X) \neq f(X^{\oplus i}))$$

For $j \neq i$, we have an inequality $\mathrm{Inf}_j(f^{\sigma_i}) \leqslant \mathrm{Inf}_j(f)$, because for $x \in \mathbf{B}^n$, if we let $A = \{x, x^{\oplus j}, x^{\oplus i}, x^{\oplus i,j}\}$ then the number of $y \in A$ with $f^{\sigma_i}(y) \neq f^{\sigma_i}(y^{\oplus j})$ is always less than or equal to the number of $y \in A$ with $f(y) \neq f(y^{\oplus j})$. In fact, the only values of $f(x), f(x^{\oplus j}), f(x^{\oplus i})$ and $f(x^{\oplus i,j})$ causing this inequality to be strict are where $f(x) = f(x^{\oplus i,j})$, $f(x^{\oplus j}) = f(x^{\oplus i})$, yet $f(x) \neq f(x^{\oplus j})$. Breaking apart $\mathbf{B}^n$ into the partitions of the elements obtained by flipping the ith and jth bits, then we obtain the influence inequality. Furthermore, we see that this inequality is tight provided not too many of these situations occur across all values of x. Similar methods show that $\mathrm{Stab}_\rho(f^{\sigma_i}) \geqslant \mathrm{Stab}_\rho(f)$. Applying this process recursively, we can always convert any Boolean function into a monotone function, increasing the stability and decreasing the influence of each variable in the process.

Another application of the Fourier expansion of influence is that there are very few Boolean valued functions with Fourier coefficients bounded by degree one. It is very similar to the Friedgut Kalai Naor theorem, but can be proved with little knowledge of the analytic properties of Boolean functions.

Lemma 2.3. *If $f : \mathbf{B}^n \to \mathbf{B}$ has $W^{\leqslant 1}(f) = 1$, then f is either constant, or $\pm x^i$ for some $i \in [n]$.*

Proof. Write $f(x) = a + \sum b_j x_j$. Then $(D_i f)(x) = b_i$, and because f is Boolean valued,

$$b_i = (D_i f)(x) = \frac{f(x^{i\to 1}) - f(x^{i\to -1})}{2} \in \{0, \pm 1\}$$

Hence $b_i = 0$ or $b_i = \pm 1$. Since $a_0^2 + \sum b_i^2 = 1$, this implies that either all of the b_i are equal to zero, in which case $a_0 = \pm 1$, or there is a single nonzero b_i with $b_i = \pm 1$. □

The **total influence** of a boolean function $f : \mathbf{B}^n \to \mathbf{B}$, 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, the bound we obtained over the influence over each coordinate implies that the total influence of the voting rule is bounded by $\sqrt{n}$, and this is asymptotically obtained by the majority function.

Example. *The total influence of the dictator x^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 pivotal for all inputs. The total influence of And_n and Or_n is $n2^{1-n}$, a rather small quantity, whereas Maj_n has total influence*

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

which is the maximum total influence of any monotone transitive symmetric function, up to a scalar multiple.

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, we can therefore see the total influence as the expected cardinality of S, where S is a random set distributed with respect 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 $\text{sens}_f(x)$ to be the number of pivotal indices for f at x.

Theorem 2.4. $\mathbf{I}(f) = \mathbf{E}[sens_f(X)]$.

Proof. We find

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

where the expectation with respect to a random, uniformly distributed index i over $[n]$. We have

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

Hence $(Lf)(x) = f(x)\text{sens}_f(x)$, and so

$$\mathbf{I}(f) = \mathbf{E}[f(X)(Lf)(X)] = \mathbf{E}[f(X)^2 \text{sens}_f(X)] = \mathbf{E}[\text{sens}_f(X)]$$

and this completes the proof. □

There is also a combinatorial interpretation of the total influence, corresponding to the geometry of the n dimensional Hamming cube. A particular Boolean-valued function $f : \mathbf{B}^n \to \mathbf{B}$ can be seen as a 2-coloring of the Hamming cube. Each vertex of the cube 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 over all edges. Since the expected number of **boundary edges** (those edges (x, y) with $f(x) \neq f(y)$) eminating from an average vertex is $\mathbf{I}(f)$, the probability that a random edge in the entire graph will be a boundary edge is $\mathbf{I}(f)/n$. We will soon find that the analysis of Boolean functions gives powerful combinatorial properties about the Hamming graph.

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 for Boolean functions: For any Boolean function f, $\mathbf{V}(f) \leqslant \mathbf{I}(f)$. The inequality is strict unless $\mathbf{W}^{>2} f = 0$. If the low degree coefficients of the function vanish, then we can give an even sharper inequality – if $\mathbf{W}^{<k}(f) = 0$, then $k\mathbf{V}(f) \leqslant \mathbf{I}(f)$. Conversely, since $\mathrm{Inf}_i(f) \leqslant \mathbf{V}(f) \leqslant \mathbf{I}(f)$, the inequality is tight when the influence is concentrated on a single voter.

Example. *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 containing m points, if we define $\alpha = m/2^n$, then the 'characteristic function'*

$$f(x) = \begin{cases} -1 & x \in A \\ +1 & x \notin A \end{cases}$$

has variation $4\alpha(1-\alpha)$, and therefore the Poincare inequality tells us that the expected number of boundary edges from the average point on the Hamming cube is lower bounded by $4\alpha(1-\alpha)$, and therefore the number of boundary edges of the set A is lower bounded by $2\alpha(1-\alpha)n2^n$. In particular, for the set A with characteristic function $x^{[n]}$, $\alpha = 1/2$, and the bound is tight, as it is for $\alpha = 0$ and $\alpha = 1$. However, for other values of α much better lower bounds can be obtained.

Lemma 2.5. *If $f : \mathbf{B}^n \to \mathbf{B}$ is unbiased, then there is an index i with $\mathrm{Inf}_i(f) \geqslant 2/n - 4/n^2$.*

Proof. Let I be the index which maximizes $\mathrm{Inf}_I(f)$ over all choices of indicies. First, note that $\mathrm{Inf}_I(f) \geqslant 1/n$, for if $\mathrm{Inf}_i(f) < 1/n$ holds for all i, then

we find $\mathbf{I}(f) < 1 = \mathbf{V}(f)$, which is impossible. Now, because f is unbiased,

$$\mathbf{I}(f) \geqslant \mathbf{W}^1(f) + 2(1 - \mathbf{W}^1(f)) = 2 - \mathbf{W}^1(f)$$

Now we use the fact that $|\widehat{f}(i)| \leqslant \mathrm{Inf}_i(f)$ to conclude that

$$\mathbf{W}^1(f) = \sum \widehat{f}(i)^2 \leqslant n\mathrm{Inf}_I(f)^2$$

hence $n\mathrm{Inf}_I(f) \geqslant \mathbf{I}(f) \geqslant 2 - n\mathrm{Inf}_I(f)^2$. Rearranging, we can calculate that $\mathrm{Inf}_I(f) \geqslant 2/n - \mathrm{Inf}_I(f)^2$, and this implies that

$$\mathrm{Inf}_i(f) \geqslant \frac{\sqrt{1 + 8/n} - 1}{2} \geqslant (2/n) - (4/n^2)$$

Thus unbiased Boolean functions give some index an influence of at least $2/n + O(1/n^2)$. □

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 2.6. *Let f be a voting rule for a 2-candidate election. If $w(x)$ is the number of indices i such that $x_i = f(x)$ (the number of voters who agree with the election 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 $f(X) = \mathrm{sgn}(\sum X^i)$, whenever $\sum X^i \neq 0$, which occurs when f is a majority function. □

Example. *Consider the function f choosen randomly from all Boolean valued functions on $\mathbf{B}^n$. 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 L_2-embeddings of the Hamming cube

Our discussion of influence can be used to give lower bounds on the ability to metrically embed the Hamming cube in Euclidean space. Recall that the Hamming cube $\mathbf{B}^n$ can be viewed as a metric space under the Hamming distance $d(x,y) = \#\{i : x_i \neq y_i\}$. Similarily, the inner product $\langle x,y\rangle = \sum x_i y_i$ on $\mathbf{R}^n$ induces a metric, with $d(x,y) = \sqrt{\sum (x_i - y_i)^2}$. If X and Y are metric spaces, we will say that $f : X \to Y$ is an embedding of metric spaces with distortion D, if $d(x,y) \leqslant d(f(x),f(y)) \leqslant Dd(x,y)$ for all $x,y \in X$. In this section we will show that every embedding of $\mathbf{B}^n$ in $\mathbf{R}^m$ has distortion at least $\sqrt{n}$.

To begin with, note that any function $f : \mathbf{B}^n \to \mathbf{R}$ can be written uniquely as the sum of an odd and even function, which we denote f^{odd} and f^{even}, which can be calculated as

$$f^{\text{odd}}(x) = \frac{f(x) - f(-x)}{2} \qquad f^{\text{even}}(x) = \frac{f(x) + f(-x)}{2}$$

If we consider a Fourier expansion $f(x) = \sum a_S x^S$, then we find

$$f^{\text{odd}}(x) = \sum_{|S| \text{ odd}} a_S x^S \qquad f^{\text{even}}(x) = \sum_{|S| \text{ even}} a_S x^S$$

Using the Fourier expansion formula for the total influence, this implies that

$$\|f^{\text{odd}}\|_2^2 = \sum_{|S| \text{odd}} a_S^2 \leqslant \sum |S| a_S^2 = \mathbf{I}(f)$$

Hence $\mathbf{E}[(f(X) - f(-X))^2] \leqslant \sum_{i=1}^n \mathbf{E}[(f(X) - f(X^{\oplus i}))^2]$. Now given any embedding $F : \{-1,1\}^n \to \mathbf{R}^m$, we obtain m coordinate functions $F = (f_1,\ldots,f_m)$. The distance formula on F tells us $\sum [f_i(x) - f_i(y)]^2 \geqslant \Delta(x,y)^2$

for all x, y. But summing up the inequality we calculated over all f_j tells us that

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\mathbf{E}[(f_j(X)-f_j(X^{\oplus i}))^2] \geqslant \mathbf{E}\left[\sum_{j=1}^{m}(f_j(X)-f_j(-X))^2\right] \geqslant n^2$$

If F has distortion D, the left side of the inequality is upper bounded by nD^2, hence $D^2 \geqslant n$, and this implies $D \geqslant \sqrt{n}$.

2.3 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 ρ-correlated, for some $\rho \in [-1,1]$ if $\mathbf{E}[XY] = \rho$. 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 ρ-correlated to X by letting $Y = X$ with probability ρ, and otherwise choosing Y uniformly randomly. We write $Y \sim N_\rho(X)$ is Y is ρ-correlated to X. This is the sense with which we will need ρ-correlation, for ρ-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 ρ-correlated to Y if the coordinates of X and Y are independant, and each coordinate in X is ρ-correlated to the corresponding coordinate in Y. We can now define the stability with respect to ρ as

$$\mathrm{Stab}_\rho(f) = \mathop{\mathbf{E}}_{Y\sim N_\rho(X)}[f(X)f(Y)]$$

where X is chosen uniformly. We of course find

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

Thus if the votes applied to some voting rule f are modified by some small random quantity ρ, the chance that this will affect the outcome of the vote is $[1 + \mathrm{Stab}_\rho(f)]/2$.

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 relation 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 $\rho \in [-1, 1]$,

$$\lim_{\substack{n\to\infty \\ n\ odd}} Stab_\rho(Maj_n) = \frac{2}{\pi}\arcsin(\rho) = 1 - \frac{2}{\pi}\arccos(\rho)$$

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

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

This shows 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 noise parameter ρ (also called the **Bonami-Beckner operator** in the literature) as

$$(T_\rho f)(x) = \mathop{\mathbf{E}}_{X\sim N_\rho(x)} f(X)$$

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

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

Hence the noise operator transforms the Fourier coeffients as

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

The connection between the noise operator and the stability is that

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

Thus stability is a quadratic form, and so we find

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

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.

That all the operators that we have found can be considered as a quadratic form should not be taken as a hint that all interesting operators in Boolean analysis are quadratic, just that the ones easiest to understand happen to be. And these operators are even more easy to understand because they are diagonalized by the characters x^S, which all happen to be Eigenvalues. And if that wasn't enough, all the eigenvalues are positive, which implies that the functionals are convex – given a functional F with $F(\sum a_S x^S) = \sum b_S a_S^2$, where $b_S \geqslant 0$, and given f and g, using the convexity of x^2 we find

$$\begin{aligned} F(\lambda f + (1-\lambda)g) &= \sum_S b_S \left[\lambda \widehat{f}(S) + (1-\lambda)\widehat{g}(S)\right]^2 \\ &\leqslant \sum_S b_S \left[\lambda \widehat{f}(S)^2 + (1-\lambda)\widehat{g}(S)^2\right] \\ &\leqslant \lambda F(f) + (1-\lambda)F(g) \end{aligned}$$

Thus these operators are very simple to work with.

Theorem 2.7. *For any $\rho, \mu \in [-1,1]$, $T_\rho \circ T_\mu = T_{\rho\mu}$.*

Proof. If X is μ correlated to the constant $x \in \mathbf{B}$, and Y is ρ correlated to X, then Y is $\rho\mu$ correlated to x, since

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

Hence we have shown $(T_\rho \circ T_\mu)(f)(x) = (T_{\rho\mu} f)(x)$ for all inputs x. From the point of Fourier analysis, we find that

$$(T_\rho \circ T_\mu)(x^S) = \mu^{|S|} T_\rho x^S = \mu^{|S|}\rho^{|S|} x^S = (\mu\rho)^{|S|} x^S$$

hence the theorem is essentially trivial to verify here, since both operators are diagonal matrices with respect to the Fourier basis. □

Theorem 2.8. *T_ρ is a contraction on $L^q\{-1,1\}^n$ for all $q \geqslant 1$.*

Proof. We can apply 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 2.9. *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

$$\mathrm{Stab}_\rho(f) = \sum_{S\neq\varnothing} \rho^{|S|}\widehat{f}(S)^2 \leqslant \rho \sum_{S\neq\varnothing} \widehat{f}(S)^2 = \rho$$

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

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

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

$$\mathrm{Stab}_\rho(f) = \mathbf{E}[f] + \rho\mathbf{W}^1(f) + O(\rho^2) \qquad \mathrm{Stab}_{1-\rho}(f) = \mathbf{V}(f) - \rho\mathbf{I}(f) + O(\rho^2)$$

We can also use these derivatives, combined with the mean value theorem from single variable calculus, to obtain a bound on how the stability of a function changes under small pertubations of the noise parameter.

Theorem 2.10. *For any Boolean function f,*

$$|Stab_\rho(f) - Stab_{\rho-\varepsilon}(f)| \leqslant \frac{\varepsilon}{1-\rho}\mathbf{V}(f)$$

Proof. Using the mean value theorem, we can find $\eta \in (\rho-\varepsilon, \rho)$ such that

$$\mathrm{Stab}_\rho(f) - \mathrm{Stab}_{\rho-\varepsilon}(f) = \varepsilon \frac{d\mathrm{Stab}_\eta(f)}{d\eta} = \varepsilon \sum k\eta^{k-1}\mathbf{W}^k(f)$$

Now we use the fact that $k\eta^{k-1} \leqslant 1/(1-\eta)$ for all $0 \leqslant \eta \leqslant 1$, hence

$$\varepsilon \sum k\eta^{k-1}\mathbf{W}^k(f) \leqslant \frac{\varepsilon}{1-\eta}\mathbf{V}(f) \leqslant \frac{\varepsilon}{1-\rho}\mathbf{V}(f)$$

Hence we have a Lipschitz inequality for $\mathrm{Stab}_\rho(f)$ which explodes as $\rho \to 1$, which makes sense because the estimate $\sum x^k = 1/(1-x)$ we used to establish this estimate breaks down at $\rho = 1$. If this inequality is sharp for some function, this means the stability of this function must drastically decrease under small variations of the noise parameter near $\rho = 1$. □

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

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

The **total** p**-stable influence** is

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

In this form, we see that $\mathbf{I}^\rho(f) = d\mathrm{Stab}_\rho(f)/d\rho$, so that this ρ-stable influence measures the change in stability of a function with respect to ρ. There isn't too much of a natural interpretation of the ρ-stable influences. Since $D_i f$ measures if f is pivotal at f, one intuitive insight about the ρ-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 ρ-stable influence now incorporates a slight change in the other coordinates as well. Regardless of their interpretation, they will be technically very useful later on.

Theorem 2.11. *Suppose that a function f satisfies $\mathbf{V}(f) \leqslant 1$. If $0 < \delta, \epsilon < 1$, then the number of 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

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

So to complete the proof, we need only prove that $k\delta(1-\delta)^{k-1} \leqslant 1$ for any $0 < \delta < 1$ and $k \geqslant 1$. Note that for $0 < x < 1$,

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

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

2.4 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 the choice of a voting system becomes unclear when we switch to a vote between three candidates, where each voter chooses an *order of preference* between the three candidates. 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 $f : \mathbf{B}^n \to \mathbf{B}$ to consider candidates pairwise, to consider such **Condorcet elections**. In this section, we consider only 3-candidate scenarios. However, proofs of the impossiblity of certain voting rules for the 3-candidate scenario will imply the impossiblity for more than three candidates, because we could specialize the voting rule on the smaller set by 'ignoring' some of the candidates.

Example. *Consider the Condorcet election over 3 candidates a, b, and c using the majority function as a voting rule. We take 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** *of the election.*

Given a particular voter, for two candidates x and y, we define the xy choice of the voter to be the element of $\mathbf{B}$, 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 $\mathbf{B}^3$, and conversely, an element of $\mathbf{B}^3$ 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 $\mathbf{B}^3$, 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 particulary 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 $\mathrm{NAE} : \mathbf{B}^3 \to \{0,1\}$. 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 2.12. *The probability of a Condorcet winner in a Condorcet election using the pairwise voting rule $f : \mathbf{B}^n \to \mathbf{B}$ is precisely*

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

Proof. Let $x, y, z \in \mathbf{B}^n$ 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

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

We therefore calculate the expectation of the operator as

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

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 2.13 (Kalai)**.** *The only voting rule $f : \mathbf{B}^n \to \mathbf{B}$ 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 2.14. *If $f : \mathbf{B} \to \mathbf{B}$ 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 2.15. *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 $f : \mathbf{B}^n \to \mathbf{B}$, 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, using some more complex theorems of the harmonic analysis.

Theorem 2.16. *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 (which says a Boolean-valued function is $O(\varepsilon)$ close to being a dictator, or the negation of a dictator, if $\mathbf{W}^1(f) \geqslant 1-\varepsilon$). □

The Friedgut-Kalai-Naor requires some sophisticated techniques in Boolean function theory, and we'll prove it in a later chapter.

2.5 The Correlation Distillation Problem

In the correlation distillation problem, a source choose a uniformly random bit string X of length n, and broadcasts it to m parties, who each recieve the bits with a certain amount of noise, obtaining slightly different random values $X_1, \ldots, X_m$. The goal is to find $\mathbf{B}$ valued functions $f_1, \ldots, f_m$ which maximize the probability that $f_1(X_1) = \cdots = f_m(X_m)$. To avoid trivial solutions, we require that $f_i(X_i)$ is unbiased. Using the theory of noise we have developed, we can find optimal strategies to the correlation distillation problem in certain situations.

The first situation we understand is the 2-party problem, where the random bit string X is distributed to the parties as independent, ρ correlated bit strings for some positive $\rho \geqslant 0$. That is $X_1, Y_1 \sim N_\rho(X)$ independently, and we must find $f, g : \mathbf{B}^n \to \mathbf{B}$ which maximizes the chance of equality. Given f and g, we consider Fourier expansions and find that

$$\begin{aligned}\mathbf{P}(f(X_1) = g(X_2)) &= \mathbf{E}\left(\frac{f(X_1)g(X_2)+1}{2}\right) \\ &= (1/2) + (1/2)\sum \widehat{f}(S)\widehat{g}(T)\mathbf{E}[X_1^S X_2^T] \\ &= (1/2) + (1/2)\sum \widehat{f}(S)\widehat{g}(S)\rho^{2|S|}\end{aligned}$$

which follows beacuse $\mathbf{E}[X_1^S X_2^T] = 0$ if $S \neq T$, and $\mathbf{E}[X_1^S X_2^S] = \rho^{2|S|}$. Now, applying the Cauchy Schwarz inequality over each subset of sets of size k, we find a bound

$$\sum \widehat{f}(S)\widehat{g}(S)\rho^{2|S|} \leqslant \sum \rho^{2k}\sqrt{\mathbf{W}^k(f)}\sqrt{\mathbf{W}^k(g)} \leqslant \rho^2$$

The first inequality turns into an equality if and only if $\sum_{|S|=k} \widehat{f}(S)\widehat{g}(S)$ is non-negative for each k, and we can find are scalars λ_k, for each degree k, such that $\widehat{f}(S) = \lambda_{|S|}\widehat{g}(S)$. The second inequality is an equality only when $\mathbf{W}^1(f) = \mathbf{W}^1(g) = 1$, which occurs only when f and g are dictators, or their negations. If both inequalities are equalities, this implies that $f = \lambda g$ for some λ, and since f and g are both Boolean valued we conclude that $f = \pm g$, and since $\langle f(X), g(X)\rangle \geqslant 0$, we must actually have $f = g$. Thus the best functions for the correlation distillation function are obtained by letting $f = g = \chi_i$ for some dictator χ_i, or letting $f = g = -\chi_i$, which has a probability of $(1+\rho^2)/2$ chance of success. Using the Friedgut-Kalai-Naor theorem, one can find that functions which have a near-optimal chance of success must be close to being a dictator or its negation, which effectively means that we should only focus on a single bit to obtain high correlation from independently noisy bits. Using similar Fourier expansion techniques, we find that this strategy is also optimal for the three party ρ correlated noise correlation distillation problem.

Now consider a second situation, where X is chosen randomly, and X_1, X_2 are obtained from X by independently letting X_1 and X_2 be equal to X with probability ρ, and then letting X_1 and X_2 by equal to 0 with probability $1-\rho$. To find functions $f : \{-1,0,1\}^n \to \mathbf{B}$ and $g : \{-1,0,1\}^n \to \mathbf{B}$ which maximize the chance of equality, TODO: FINISH HERE.

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. If $f : \mathbf{B}^n \to \mathbf{R}$, and $\mathcal{T}$ is a family of subsets of $[n]$, then we say f is ε-**concentrated** on $\mathcal{T}$ if

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

In particular, we say a function is ε-concentrated on degree n if f is ε concentrated on the subsets of size less than or equal to n, which also be written as saying that $\mathbf{W}^{\geqslant n}(f) \leqslant \varepsilon$. For Boolean valued functions, ε-concentration on $\mathcal{T}$ is equivalent to saying that $\mathbf{P}(|S| \notin \mathcal{T}) \leqslant \varepsilon$, where S has the induced spectral distribution.

Because the total influence of a function is obtained by amplifying high degree Fourier coefficients, we can obtain initial bounds on spectral concentration in terms of the function's total influence.

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

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 $O(2^{-n/\log n})$, hence the function is ε concentrated on degree up to $O(2^{-n/\log n}\varepsilon^{-1})$. This means that the Tribes function is concentrated on an incredibly small degree for large values of n. The Tribes function effectively separates the correlation between large groups of indices, which implies that the function should be distributed mainly on low degree coefficients.*

Another estimate of the spectral concentration is obtained from the noise stability of a function, since the noise stability amplifies a coefficient of size k by a coefficient $1-(1-2\delta)^k$, and if a function is not concentrated on low degree coefficients, we can lower bound the modification.

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. *Recall that the majority function has noise sensitivity*

$$\mathbf{NS}_\delta(Maj_n) = \frac{2\sqrt{\delta}}{\pi} + O(\delta^{3/2}) + o(1) = O(\sqrt{\delta}) + o(1)$$

In fact, if we take n large enough, and δ sufficiently small, we find that $\mathbf{NS}_\delta(Maj_n) \leqslant \sqrt{\delta}$*, hence* Maj_n *is* $3\sqrt{\delta}$ *concentrated on degree m for* $m \geqslant 1/\delta$*, or that* Maj_n *is* ε *concentrated on degree m for* $m \geqslant 9/\varepsilon^2$*. This also shows that the first concentration bound we found is not always tight, because*

$$\mathbf{I}(Maj_n) = \sqrt{\frac{2n}{\pi}} + o(1)$$

and the first bound only gives that Maj_n *is* ε *concentrated on degree* $O(\sqrt{n})/\varepsilon$*.*

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 functions, we can obtain bounds on the number of coordinates relevant to the definition of 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 $f(-1)$ is non-zero, so the lemma is satisfied here. In general, if we write

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

and 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. □

Corollary 3.4. *If f is a Boolean-valued function of degree less than or equal to k, and $Inf_i(f) \neq 0$, then $Inf_i(f) \geqslant 2^{1-k}$.*

Proof. If f has degree $\leqslant k$, then $D_i f$ has degree $\leqslant k-1$, so if $D_i f \neq 0$ then $\mathbf{P}((D_i f)(X) \neq 0) \geqslant 2^{1-k}$, and this is exactly the inequality we wanted. □

Theorem 3.5. *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 $\mathbf{I}(f)/2^{1-k}$. Since

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

This means that $\mathbf{I}(f)/2^{1-k} \leqslant k2^{k-1}$, and so f is a $k2^{k-1}$ junta, because a coordinate i is relevant to the function f if and only if $\mathrm{Inf}_i(f) > 0$. □

The Friedgut Kalai Naor theorem is essentially a more robust version of this theorem, but can only address the case where $k = 1$. The next section gives us a reason why we can't improve this theorem by much more, because decision trees provide us with a method of constructing functions f with degree less than or equal to k which can access $k2^{k-1}$ coordinates of the function.

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 trees 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 3.6. *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 $a_P^S \in \mathbf{B}$, 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 3.7. *If $f : \mathbf{B}^n \to \mathbf{B}$ is computable by a decision tree of size s, and $0 < \varepsilon \leqslant 1$, then the spectrum of f is 2ε 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 $\varepsilon/2$-close to f, because g only differs from f on the paths of length $l \geqslant \log(s/\varepsilon)$. Since each query the decision tree makes subdivides the domain in two, these paths only determine the values of $2^{-l} \leqslant 2^{-\log(s/\varepsilon)} = \varepsilon/s$ elements of the domain. By choosing the value of the leaf of T_0 to be equal to be the majority of the values over the sublets in T, we can actual obtain that we change less than or equal to $\varepsilon/2s$ values of the leaf. Since there are only s paths in total, we can only change at most $\varepsilon/2$ elements of the domain. Since T_0 has depth less than $\log(s/\varepsilon)$, g has degree less than $\log(s/\varepsilon)$, so it follows that if $m \geqslant \log(s/\varepsilon)$, then

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

and this completes the proof. □

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, so we may apply our current theory, that we are trying to determine Boolean-valued functions. The two models of obtaining data in order to make a decision are

- **random examples**, where we draw $(X, f(X))$ from a uniform distribution over the variable X.

- **queries**, where we can request a specific value x for any $x \in \mathbf{B}^n$.

The query method is at least as powerful as the random example method, because we can always randomly select samples ourselves. After a certain number of data collection steps, we are required to choose a **hypothesis function**. We say that an algorithm **learns** $\mathcal{C}$ **with error** ε if for any $f \in \mathcal{C}$, the algorithm outputs a hypothesis function (not necessarily in $\mathcal{C}$) which is ε-close to f with high probability. The standard is to output a ε-close function with probability greater than or equal to $1/2$.

A simple theory about hypothesis testing is that the more 'simple' the class $\mathcal{C}$ is, the faster it should be to recognize a hypothesis function. On the other hand, if $\mathcal{C}$ is suitably complex, then exponential running time may be necessary. The main result about this section is that learning a function is equivalent to learning it's Fourier spectra, so that functions concentrated on a sufficiently small Fourier set are easy to learn.

Theorem 3.8. *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 3.9. *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 3.10. *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$(|\mathcal{F}|, n, 1/\delta)$ time, we can with high probability produce a hypothesis function $\varepsilon + \delta$ 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 $\log(1/\rho) \cdot \text{poly}(n, 1/\delta)$ 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 $h(x) = \text{sgn}(g(x))$. 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 $\log(1/\rho) \cdot \text{poly}(n, 1/\delta)$ time. If we let $\delta = \sqrt{\delta/|\mathcal{F}|}$, we find h is $\delta + \varepsilon$ close to f, in time $\log(1/\rho) \cdot \text{poly}(n, |\mathcal{F}|, 1/\delta)$. □

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 $\log(1/\rho) \cdot poly(n^{\log(s/\varepsilon)}, 1/c) = \log(1/\rho) \cdot poly(s/\varepsilon, 1/c)$.*

The Goldreich Levin theorem provides a general method for finding a family $\mathcal{F}$ upon which a general function is ε concentrated, and provides the final component for learning over an arbitrary family.

3.4 Walsh-Fourier Analysis Over Vector Spaces

It will be helpful to introduce some notation before attacking the Goldreich Levin theorem. Note that each $\{0,1\}^n$ is a vector space over the field $\{0,1\}$. What's more, we have a map $\langle x, y\rangle = (-1)^{\sum x_i y_i}$ which is 'bilinear', in the sense that

$$\langle x + y, z\rangle = \langle x, z\rangle\langle y, z\rangle \qquad \langle x, y + z\rangle = \langle x, y\rangle\langle x, z\rangle$$

so the map is a homomorphism from $\{0,1\}^n$ to $\mathbf{B}$ for each fixed variable. Each $x \in \{0,1\}^n$ then gives rise to a character x^* defined by $x^*(y) = \langle x, y\rangle$, and $(x + y)^* = x^* y^*$, so the map is a homomorphism. Since the bilinear form $(x, y) \mapsto \sum x_i y_i$ is non-degenerate, the map $x \mapsto x^*$ is injective, and since the character group has the same cardinality as the group, the homomorphism is actually an isomorphism; every character on $\{0,1\}^n$ must be represented by x^* for some x. In fact, if $\chi_S : \{0,1\}^n \to \mathbf{B}$ is a character, then the corresponding x^* is found by letting x be the $\{0,1\}$ indicator

function corresponding to S, letting $x_i = 1$ if $i \in S$, and $x_i = 0$ otherwise. Because of this correspondence, we will write $\widehat{f}(x)$ for the corresponding $\widehat{f}(S)$, which can be defined as

$$\widehat{f}(x) = \mathbf{E}[f(X)x^*(X)] = \mathbf{E}[f(X)\langle x, X\rangle]$$

and this is a definition invariant of the original Fourier coefficients.

Given a subspace V of $\{0,1\}^n$, we let $V^\perp$ be the subspace of elements x of $\{0,1\}^n$ such that $\langle x, y\rangle = 1$ for all $y \in V$. By general properties of non-degenerate bilinear forms, $(V^\perp)^\perp = V$, and we can write $\{0,1\}^n = V \oplus V^\perp$, so $V^\perp$ has dimension $n - m$.

Theorem 3.11. *If V is a subspace of $\{0,1\}^n$ of codimension m, then*

$$\mathbf{I}(x \in V) = \frac{1}{2^m} \sum_{y \in V^\perp} y^*$$

Proof. Since $(V^\perp)^\perp = V$, a vector x is in V if and only if $y^*(x) = 1$ for all $y \in V^\perp$. Let $y_1, \ldots, y_m$ form a basis for $V^\perp$. Then

$$\begin{aligned}\mathbf{I}(x \in V) &= \frac{1}{2^m} \prod_{y=1}^{m} \left(y_i^*(x) + 1\right) = \frac{1}{2^m} \sum_S \left(\prod_{i \in S} y_i^*(x)\right) \\ &= \frac{1}{2^m} \sum_S \left(\sum_{i \in S} y_i\right)^*(x) = \frac{1}{2^m} \sum_{y \in V^\perp} y^*(x)\end{aligned}$$

This gives the required Fourier expansion. □

More generally, if V is an affine subspace of $\{0,1\}^n$, with $V = x + W$ for $x \in V$. We can describe it equivalently as the set of points y such that for any $z \in W^\perp$, $z^*(y) = z^*(x)$. Then if W has codimension m, then

$$\mathbf{I}(y \in V) = \mathbf{I}(y + x \in W) = \frac{1}{2^m} \sum_{z \in W^\perp} z^*(y + x) = \frac{1}{2^m} \sum_{z \in W^\perp} z^*(x) z^*(y)$$

Thus the indicator has sparsity 2^m, and is 2^{-m} granular.

Given $f : \mathbf{B}^n \to \mathbf{R}$, it will be more natural to think of the function as $f : \mathbf{B}^{[n]} \to \mathbf{R}$, i.e. the function which takes an input $x : [n] \to \mathbf{B}$, and outputs

a number $f(x)$. We can consider the Fourier expansion of an arbitrary $f : \mathbf{B}^A \to \mathbf{R}$, 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 $z \in \mathbf{B}^J$, we can define $f_{J|z} : \mathbf{B}^{J^c} \to \mathbf{R}$ to be the function obtained by fixing all indices of J with the values z. It follows that if $S \subset J^c$, then

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

Picking z randomly over all choices, we find

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

and

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

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

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

which is the weight over irrelevant indices. A crucial tool of Golreich-Levin is that this is the expected value of $\widehat{f_{J|Z}}(S)^2$, so that we can estimate this expected value using the law of large numbers by picking particular values of Z, and use Chernoff bounds to open precise estimates of how close our estimates will be to the mean.

3.5 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 $f : \mathbf{B}^n \to \mathbf{B}$, 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.

Theorem 3.12. *For any $S \subset J \subset [n]$, an algorithm with query access to $f : \mathbf{B}^n \to \mathbf{B}$ 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

The PCP Theorem, and Probabilistic Proofs

The Cook-Levin discovery of NP completeness hints that there are a certain class of **NP** problems which are computationally much more complex than other problems. In particular, the theorem says that if the Boolean satisfiability problem SAT has a polynomial time algorithm, then all problems in **NP** are verifiable in polynomial time, solving the $\mathbf{P} = \mathbf{NP}$ conjecture in the affirmative. Since most computer scientists do not believe that $\mathbf{P} = \mathbf{NP}$, we expect SAT not to have a polynomial time solution. In the face of this computational wall, it is natural to explore other strategies to 'solving' the satisfication problem.

Rather than considering an algorithm which is completely correct, we instead try to find algorithms which are 'approximately correct'. Consider the MAX 3-SAT problem, in which given a 3-SAT instance x, we are asked to find a truth assignment which maximizes the fraction of clauses that can be satisfied. We call this the value of x, denoted $\text{val}(x)$. We say an algorithm is a ρ-approximation for MAX-3SAT if the output of the algorithm given an input x consisting of m clauses is a truth assignment satisfying at least $\rho \cdot m\text{val}(x)$ of the clauses. It is an interesting question to ask whether there is a boundary in how well MAX-3SAT can be approximated. That is, are there polynomial time algorithms which can approximate MAX-3SAT to an arbitrary degree of precision?

The PCP theorem essentially answers this in the negative. In short, the theorem shows that there is $\rho < 1$ such that for every language $L \in \mathbf{NP}$, there is a polynomial-time computable function f mapping strings in the

alphabet of L to $3SAT$ instances such that if $x \in L$, then $\text{val}(f(x)) = 1$, and if $x \notin L$, then $\text{val}(f(x)) < \rho$. Thus the PCP theorem immediately implies that if there are ρ-approximation algorithms for MAX-3SAT for ρ arbitrarily close to 1, then $\mathbf{P} = \mathbf{NP}$. Thus the theorem essentially guarantees the existence of problems which are 'hard to approximate'.

Is is natural to try and adapt the standard proof of the Cook-Levin theorem to proving that this claim is true. After the proof of Cook's theorem in the 1970s, researchers attempted to use the reduction to separate functions by value, but they found these methods did not suffice; Cook's reduction yields polynomial-time computable functions f mapping strings over an alphabet representing an NP problem L into a Boolean function such that $\text{val}(f(x)) = 1$ if and only if $x \in L$, but the construction does not create a 'gap' in the values of $f(x)$ for $x \notin L$ – indeed for most problems we find that there are values $\text{val}(f(x))$ which are arbitrarily close to 1, with $x \notin L$. The PCP theorem therefore finds a truly novel encoding of problems as Boolean satisfaction problems, giving a result that MAX-3SAT is difficult to approximate.

4.1 PCP and Proofs

There is another way to look at the PCP theorem from the perspective of formal computability theory. Recall that the class **NP** consists of problems whose solutions can be verified in polynomial time. That is, a language L is in **NP** if there is a Turing machine taking an input x and certificate π that runs in time polynomial to the input x, and such that $x \in L$ if and only if the turing machine accepts an input (x, π) for some certificate π. If we see π as a 'proof' of the validity of x, then an NP problem is one whose proofs can be effectively checked.

Suppose that we weaken this condition, considering randomized algorithms which reject invalid proofs with high probability. We would expect that these algorithms form a class much more general than **NP**; the PCP theorem says that if we restrict access to the verification certificate, and to the amount of randomness that a function can have, then this new classification of problems is no more general than **NP**. This also implies that certificates to problems in **NP** can be encoded in a way that solutions require little access to the certificate. Thus the theorem gives bounds on the amount of randomness and the amount of checking required to dis-

tinguish certain encodings of **NP** problems.

If we are to restrict a machines access to the certificate π, we must distinguish between the types of access a machine could have. **Non-adaptive** queries are made based only on the input x, and perhaps some randomness, whereas **adaptive queries** are allowed to depend on previous inputs. We will restrict ourselves to non-adaptive queries, because these have effective derandomization properties, but the question of allowing adaptive queries is still interesting.

So let's define the notion of a randomized certificate checker. Given a language L, and functions $f, g : \mathbf{N} \to \mathbf{N}$ we define a (f, g) **PCP verifier** of L to be a probabilistic Turing machine, running in polynomial time in x which, given an input x of length n, and given random access to a certificate π (called the 'proof' of x), accepts or rejects x based on at most $g(n)$ random bits and $f(n)$ non-adaptive queries to locations of π. We require that if $x \in L$, then there is a proof π such that the machine is certain to accept x, in which case we call π the **correct proof**, or if $x \notin L$, then for *every* proof π, the machine rejects x with probability greater than $1/2$. We define the language $\mathbf{PCP}(f, g)$ to be the space of problems which have a (f, g) verifier. More generally, for classes C, C' of functions (like $O(n)$, $\mathrm{poly}(n)$, etc), we define the language $\mathbf{PCP}(C, C')$ to be the class of problem with an (f, g) verifier, where $f \in C$ and $g \in C'$.

We note that the probability with which a false x is rejected can be set to any constant value. It is just set to $1/2$ customarily. The reason for this generality is that if a problem L has a (f, g) verifier rejecting every $x \notin L$ with probability $1 - P$, and by running this verifier independently n times, we obtain a (nf, ng) verifier which rejects x with probability $1 - P^n$, and has the same asymptotic querying properties. Furthermore, we might as well assume that the length of any certificate π we consider has length $f(n)2^{g(n)}$, because an algorithm using k random bits to choose l non-adaptive queries can only choose between $l2^k$ possible locations to query from.

Though PCP verifiers are random, they can often be derandomized with an exponential time increase. If we have a PCP verifier for a problem L, using $f(n)$ points of access to the certificate, and $g(n)$ random bits, then we may design a Turning machine which, given x and π, simulates the PCP verifier on all $2^{g(n)}$ possible random inputs, calculates the probability of accepting x, and rejects x if the probability that x is rejected is greater than $1/2$. This Turing machine verifies L precisely, and runs in

time $O(\max(2^{g(n)+O(\log n)}), \text{poly}(n))$.

We can guess π by nondeterministically choosing $f(n)2^{g(n)}$ symbols for π, and then running the computer on all these inputs. Thus if L is an (f,g) verifiable problem, then L is also a problem in $\mathbf{NTIME}(f(n)2^{g(n)+O(\log n)})$, the space of problems solvable by a nondeterministic Turing machine in a specified amount of time. In particular, this implies that any problem in $\mathbf{PCP}(O(\log n), O(1))$ can be solved by a nondeterministic Turing machine running in time $O(\log n)2^{O(\log n)}$, hence $\mathbf{PCP}(\log n, 1) \subset \mathbf{NP}$. The PCP theorem is the converse to this statement.

Theorem 4.1 (The PCP Theorem)**.** $\mathbf{NP} = \mathbf{PCP}(O(\log n), O(1))$

We therefore obtain the surprising fact that there is a particular encoding of the certificates of **NP** problems which enable the problem to be verified with high probability restricting an algorithm to logorithmic randomness, and only checking a constant number of symbols in some proof of a solution. This leads to surprising results. For instance, given an axiomatic system in which proofs can be verified in polynomial time to the size of the proof, the language

$$\{\langle x, 1^n \rangle : x \text{ is a formula with a proof of length at most } n \text{ characters}\}$$

is in **NP**. It follows that we can encode the proofs in this axiom system in such a way that we can check the correctness of an encoded proof in polynomial time with randomness linear in the size of the proof, and only looking with a constant number of bits! Though we rarely know how long a proof should be before we find a proof, hardly any proofs have been discovered with a length greater than 10^{10} characters, so we should expect a PCP verifier for this language.

Example. *Consider the graph non-isomorphism problem, which asks us to determine, given a pair of graphs $\langle G_0, G_1 \rangle$, each containing n nodes, whether G_0 is not isomorphic to G_1. We claim the problem is in $\mathbf{PCP}(poly(n), O(1))$. First, we index all graphs with n nodes by an integer between 1 and 2^{n^2}, so that certificates π of length 2^{n^2} can be considered as Boolean valued maps from the set of all graphs with n nodes. A 'true' proof π that G_0 is not isomorphic to G_1 shall let $\pi(X) = 0$ if X is isomorphic to G_0, $\pi(X) = 1$ if X is isomorphic to G_1, and $\pi(X)$ chosen arbitrarily if neither holds. Our verifier picks an index $i \in \{0,1\}$, and a permutation $\nu \in S_n$, uniformly at random, using $1 + \log(n!) = O(n \log n)$*

random bits. We use ν to rearrange the vertices of G_i to obtain an isomorphic graph $\nu(G_i)$. We accept if $\pi(\nu(G_i)) = i$. If we have a correct proof π, and G_0 is not isomorphic to G_1, then the verifier always accepts, as required. If G_0 is isomorphic to G_1, then the probability distribution of $\nu(G_i)$ is independent of i, and therefore $\mathbf{P}(\pi(\nu(G_i)) = i) = 1/2$, hence the verifier rejects the instance with probability $1/2$.

It turns out that $\mathbf{PCP}(\text{poly}(n), O(1)) = \mathbf{NEXP}$, so that the verifier construction above is a special case of a more general theorem. However this theorem requires a sophisticated construction, which we won't prove here.

4.2 Equivalence of Proofs and Approximation

The two interpretations of the PCP theorem are, of course, equivalent. This becomes clear when we introduce the family of k constraint satisfaction problems, denoted k-CSP. In the problem, we take a set of k-juntas $f_1, \ldots, f_m : \{0,1\}^n \to \{0,1\}$, called the constraints. An assignment $x \in \{0,1\}^n$ satisfies the constraints if $f_i(x) = 1$ for each i. In general, we define the value of the constraints f_i to be the maximum fraction of the constraints that can be satisfied, denoted $\text{val}(f)$. The 3-SAT problem is a subproblem of the 3-CSP problem, where each f_i is a disjunction of variables.

Now given a natural number k and $\rho \leqslant 1$, define the YES ρ-GAP k-CSP problem to be the problem of determining, given an instance f of k-CSP, whether $\text{val}(f) = 1$, and the NO ρ-GAP k-CSP problem to determine whether $\text{val}(f) < \rho$. We say ρ-GAP k-CSP is **NP**-hard if, for any **NP** problem L, there is a polynomial time function g mapping strings in L to strings for the ρ-GAP k-CSP problem, such that if $x \in L$, then $\text{val}(g(x)) = 1$, and if $x \notin L$, then $\text{val}(g(x)) \leqslant \rho$. There is a natural equivalent statement of the PCP theorem in the context of k-CSP problems.

Theorem 4.2. *There is k and $\rho \in (0,1)$ such that ρ-GAP k-CSP is* **NP** *hard.*

To see the equivalence, suppose that $\mathbf{NP} = \mathbf{PCP}(O(\log n), O(1))$. We will show $1/2$-GAP k-CSP is **NP** hard for some k. It is sufficient to reduce 3-SAT to a problem of this type, because we can reduce all problems in **NP** to 3-SAT in polynomial time. Because of the assumption, there is a PCP verifier for 3-SAT using $K \log n$ random bits and K' queries to the certificate. Given any input x to 3-SAT and $r \in \{0,1\}^{K \log n}$, let f_{xr} be the function

that takes as input a proof π, and outputs 1 if the verifier accepts π on input x and random bits r. Note that f_{xr} only depends on at most K' entries of π. For every $x \in \{0,1\}^n$, the collection of all f_{xr} is describable with $K'n^K$ bits, and since each f_{xr} runs in polynomial time, we can transform an input x to the set of all f_{xr} in polynomial time in the size of x. If x is satisfiable, then $\text{val}(x) = 1$, whereas if x is not satisfiable, then $\text{val}(x) \leqslant 1/2$. Thus we have reduced 3-SAT to $1/2$-GAP K'-CSP in polynomial time, and therefore the $1/2$-GAP K'-CSP problem is **NP** hard.

Conversely, suppose ρ-GAP k-CSP is **NP** hard for some ρ and k. Given any **NP** problem L, consider a reduction g to ρ-GAP k-CSP with constraints f. Consider the following PCP verifier for an instance of k-CSP. We shall interpret a certificate π as an assignment to the variables x_i such that all constraints are satisfied. Our algorithm will take a random index i, and check whether $f_i(x) = 1$ (we need only access k bits of π to determine this). If $x \in L$, then $g(x)$ will always be accepted when π is the correct proof. If $x \notin L$, then $\text{val}(g(x)) < \rho$, hence the probability of $g(x)$ being accepted by the verifier is less than ρ. This implies that L is in $\mathbf{PCP}(O(\log n), O(1))$.

Now suppose that 3-SAT is hard to approximate, for $\rho > 0$, where for every $L \in \mathbf{NP}$, there is a reduction f of L to 3-SAT such that $\text{val}(f(x)) = 1$ if $x \in L$, and $\text{val}(f(x)) < \rho$ if $x \notin L$. This is essentially a special case that ρ-GAP 3-CSP is **NP** hard. To complete the equivalence, it suffices to verify that the **NP** hardness of ρ-GAP k-CSP reduces to the **NP** hardness of the more general ρ'-GAP 3-CSP problem, for perhaps a slightly relaxed $\rho' > \rho$. If $\rho = 1 - \varepsilon$, let $f_1, \ldots, f_n$ be a k-CSP instance with n variables and m constraints. Because each f_i is a k-junta, it can be expressed as the conjunction of at most 2^k clauses, where each clause is the or of at most k variables or their negations. Let X be the set of all such constraints, bounded in size by $m2^k$. If f is a YES instance of ρ-GAP k-CSP, then there is a truth assignment satisfying all of the clauses of X. If f is a NO instance, then every assignment violates at least ε of the conditions f_i, and therefore $\varepsilon/2^k$ of the clauses of X. Using the Cook-Levin technique, we can transform any clause C on k variables to k clauses $C_1, \ldots, C_k$ over the variables $x_1, \ldots, x_k$ with additional variables $y_1, \ldots, y_k$ such that each C_i has only 3 variables, and if the x_i satisfy C, then we may pick y_i to satisfy all C_i simultaneously, and if the x_i cannot satisfy C, then for every choice of y_i some C_j is not satisfied. Let Y denote the 3-SAT instance of $km2^k$ clauses over the new

set of $n + km2^k$ variables obtained from X. If X is satisfiable, so is Y, hence $\text{val}(Y) = 1$, and if X is not satisfiable, then every assignment verifies at least $\varepsilon/k2^k$ of the constractions of Y. Thus $\text{val}(Y) \leqslant 1 - \varepsilon/k2^k$, and thus we have reduced L to MAX 3-SAT with a $1 - \varepsilon/k2^k$ separation.

4.3 NP hardness of approximation

As the Cook-Levin theorem gives us a whole family of NP-complete problems, the PCP theorem gives us hardness of approximation results for many more problems than 3SAT and CSP. As an example, we consider the approximation problem for the MAX-INDSET problem, which asks us to find the maximum independent set in a grpah (the largest number of vertices sharing no edge), and the MIN-COVER problem, which asks us to find the minimum set of vertices on a graph such that every vertex in the graph is connected by an edge to this set.

The complement of every independent set is a cover, and the complement of every cover is an independent set, so that MIN-COVER is equivalent to MAX-INDSET as exact problem classes. This does not extend to a more robust equivalence. If C is the size of the minimum vertex cover, and I the size of the maximum independent set, then $C + I = n$. Therefore, if we have a ρ approximation algorithm for MAX-INDSET, returning a set S with $\rho I \leqslant S$, then we would find a corresponding cover of size

$$n - S \leqslant \frac{n - \rho I}{n - I}(n - I) = \frac{n - \rho I}{C} C$$

and we therefore we have an approximation algorithm for the MIN-COVER problem of ratio $(n - \rho I)/C$. A similar relationship exists between transforming approximation algorithms of MIN-COVER to MAX-INDSET. However, it turns out that the approximation ratios of these two problems are very different – there is no constant-value approximation algorithm for the MAX-INDSET problem, whereas there is a trivial $1/2$ approximation for MIN-VERTEX-COVER.

Lemma 4.3. *There is a polynomial time reduction f from 3-SAT formulas containing n clauses to graphs such that $f(X)$ is an $7n$-vertex graph whose largest independant set is $n \cdot val(X)$.*

Proof. Consider the standard exact reduction of 3-SAT to independent set, which takes a logical equation X with m clauses, and returns a graph $f(X)$ with $7m$ vertices, such that there is an assignment satisfying k clauses of X if and only if $f(X)$ has an independent set of size k. We do this by associating with each clause C seven nodes of the form C_{xyz}, where $(x, y, z) \in \{0, 1\}^3$ is a truth assignment of the variables in the clause which satisfy the clause (there are 7 total possible assignments). Given two clauses C^1 and C^2, put an edge between $C^1_{x_0y_0z_0}$ and $C^2_{x_1y_1z_1}$ if the assignments are incompatible. Certainly all 7 nodes connected to a single clause are connected. An independent set in this graph of size k is then a set of consistent assignments satisfying at least k clauses, and conversely, a consistent satisfaction of k clauses leads to an independent set in the graph of size k. □

Theorem 4.4. *There is $\gamma < 1$ and $\lambda > 1$ such that if there is a γ-approximation algorithm for MAX-INDSET, or a λ approximation algorihtm for MIN-COVER, then* $\mathbf{P} = \mathbf{NP}$.

Proof. Let L be an **NP** language. The PCP theorem implies there is a polynomial time computable reduction f to MAX-3SAT. That is, for some $\rho < 1$, X is satisfiable and $\text{val}(f(X)) = 1$, or X is not satisfiable and $\text{val}(f(X)) < \rho$. The last theorem implies that if we had a ρ-approximation to IND-SET, then we could do a ρ-approximation on MAX-3SAT, thereby proving $\mathbf{P} = \mathbf{NP}$. For MIN-VERTEX-COVER, the minimum vertex cover of the graph obtained in the previous lemma has size $n[1 - \text{val}(X)/7]$. Hence if MIN-VERTEX-COVER had a ρ' approximation algorithm for $\rho' = (7-\rho)/6$, then we could find a vertex cover of size $\leqslant \rho' n(1 - 1/7) \leqslant n(1 - \rho/7)$ if $\text{val}(X) = 1$, hence we could find a MAX-IND-SET of size greater than $n\rho/7$, and this shows for ρ close enough to 1, a ρ' approximation to MIN-VERTEX-COVER would imply $\mathbf{NP} = \mathbf{P}$. □

It turns out that MAX-INDEPENDENT-SET has an 'amplification procedure', which can generate constant time approximation algorithms of arbitary accuracy, given the existence of a particular solution. This is given by the 'graph product' technique. Given a graph G with n nodes,let G^k be the graph on $\binom{n}{k}$ vertices corresponding to subset of nodes in G of size k. Define two subsets S and T to be adjacent if $S \cup T$ is an independent set. The largest independent set in G^k consists of all k-size subsets of the largest independent set in G, and therefore has size $\binom{I}{k}$, where I is the

maximum independent set in G^k. Thus if a ρ approximation algorithm for the MAX-INDEPENDENT-SET problem implies $\mathbf{P} = \mathbf{NP}$, then a

$$\frac{\binom{\rho I}{k}}{\binom{I}{k}} = \frac{(\rho I)(\rho I - 1)\ldots(\rho I - k + 1)}{I(I-1)\ldots(I-k+1)} = \rho^k \prod_{m=0}^{k-1} \frac{I - m/\rho}{I - m} \leqslant \rho^k$$

approximation algorithm for k powers of graphs implies $\mathbf{P} = \mathbf{NP}$. For k large enough, we find that the existence of any constant time algorithm for MAX-INDEPENDENT-SET implies $\mathbf{P} = \mathbf{NP}$.

4.4 A Weak Form of the PCP Theorem

We shall now prove a weak form of the PCP theorem, which can be used to attack the finer problem. First, we consider the notion of Walsh-Hadamard encoding. Given a string $x \in \mathbf{F}_2^n$, we define the **Walsh-Hadamard** encoding to be the string $W(x) : 2^{[n]} \to \mathbf{F}_2$ such that $W(x)(S) = \sum_{i \in S} x_i$. The encoding is essentially the tuple representation of $x^* : \mathbf{F}_2^n \to \mathbf{F}_2$, where $x^*(y) = \sum x_i y_i$. The encoding is very inefficient, since elements of $\mathbf{F}_2^n$ are length n bit strings, but we shall find the encoding is very useful to represent proof certificates of certain decision problems over the field $\mathbf{F}_2$, because it enables us to compute the inner product of x and y with access to a single bit of the proof certificate.

Theorem 4.5. $\mathbf{NP} \subset \mathbf{PCP}(poly(n), O(1))$.

Proof. We will consider a particular **NP** complete problem, and show it has a verifier which is $(\text{poly}(n), O(1))$. Since every **NP** is exactly reducible to this **NP** complete problem, this suffices to prove the full inclusion. The **NP** complete problem we will use is the problem QUAD-SAT, which asks to solve solve a system of m quadratic equations over $\mathbf{F}_2^n$. This problem is NP complete – it is most easiest to see this by reducing the problem of $CKT-SAT$, determing whether a Boolean circuit is satisfiable, expressing AND and OR operations by quadratic polynomials. We may assume that each term of the system contains a pair of terms, since $x^2 = x$ in $\mathbf{F}_2$. If the equations are

$$\sum a_{ij}^k x_i x_j = b_k$$

where k ranges from 1 to m, then we see that we can encode the problem as a '$m \times n^2$' matrix A, and $b \in \mathbf{F}_2^m$. If we consider the n^2 dimensional vector $(x \otimes x)_{ij} = x_i x_j$, for $x \in \mathbf{F}_2^n$, then we can write the equation as $A(x \otimes x) = b$.

Now suppose that an instance (A, b) of QUAD-SAT is satisfiable by some vector x. We will interpret a proof π as a pair of functions

$$f : \mathbf{F}_2^n \to \mathbf{F}_2 \qquad g : \mathbf{F}_2^{n \times n} \to \mathbf{F}_2$$

where f is the Walsh-Hadamard encoding of x, and g is the Walsh Hadamard encoding of $x \otimes x$. The encoding enables us to calculate values $\sum_{i \in S} x_i$ and $\sum_{(i,j) \in S} x_i x_j$ with only a single query to the certificate. The proof consists of a certain number of steps

1. First, we use the certificate to verify that f and g are linear. Otherwise, the proof is not correctly encoded. We random choose whether to test f and g, and then using three queries, we test either of the functions. If f is ε far away from being linear, or g is ε far away from being linear, the test is rejected with probability $(1 - \varepsilon)/2$. Otherwise, we may assume from now on that f and g are ε-close to being linear, and we may use local decoding to calculate values of the linear function f and g approximate with high probability. Thus let $d(f, \tilde{f}) < \varepsilon$, and $d(g, \tilde{g}) < \varepsilon$. Since $\tilde{f}$ and $\tilde{g}$ are linear, they actually encode elements of $\{0,1\}^n$ and $\{0,1\}^{n^2}$ respectively.

2. Check if $\tilde{g}$ encodes $x \otimes x$, where $\tilde{f}$ encodes x. To do this, if the equality held, then we would find

$$\tilde{f}(y)\tilde{f}(z) = \langle x, y \rangle \langle x, z \rangle = \sum_{ij} x_i x_j y_i z_j = \langle x \otimes x, y \otimes z \rangle = \tilde{g}(y \otimes z)$$

   If we choose $Y, Z \in \{0,1\}^n$ uniformly at random, and check whether $\tilde{f}(Y)\tilde{f}(Z) = \tilde{g}(Y \otimes Z)$ using 9 queries of the certificate. If $\tilde{g}$ does not encode $x \otimes x$, then $x \otimes x - \tilde{g}$ is a non-zero linear functional, hence it takes value 0 on half the inputs, and value 1 on the other half, so

$$\mathbf{P}(\tilde{f}(Y)\tilde{f}(Z) = \tilde{g}(Y \otimes Z)) = 1/2$$

   Assuming f is ε close to $\tilde{f}$, and g is ε close to $\tilde{g}$, we calculate $\tilde{f}$ and $\tilde{g}$ correctly with probability $1 - 2\varepsilon$ each. Thus we find that if $\tilde{g}$ does not encode $x \otimes x$, then we reject with probability at least $(1 - 2\varepsilon)^2/2$.

3. All that remains is to check that $\tilde{g}(A^k) = b_k$ for each k. To do this for each k would take far too many queries to the certificate. Thus we pick a random $z \in \mathbf{F}_2^n$, and then determine if

$$\langle z, \tilde{g}(A) \rangle = \sum z_i \tilde{g}(A_i) = \sum z_i b_i = \langle z, b \rangle$$

If $\tilde{g}(A) \neq b$, then provided we pick Z uniformly $\langle Z, \tilde{g}(A) \rangle \neq \langle Z, b \rangle$ with probability $1/2$, and therefore we reject with probability at least $(1-2\varepsilon)/2$, with only four queries to the certificate.

In conclusion, with 16 queries of the certificate, we reject an incorrect proof with probability greater than or equal to

$$\min\left(1-2\varepsilon, \frac{1-2\varepsilon}{2}, \frac{(1-2\varepsilon)^2}{2}\right) = \frac{(1-2\varepsilon)^2}{2}$$

And, with a more stringent rejection probability (randomly performing one of the steps above, not all of them in sequence), we can perform the test with only 9 queries. □

4.5 Property Testing

For intuition, we now discuss the field of property testing, which has an intrinsic connection to probabilistically checkable proofs. In particular, we look at dictatorship testing, which is essentially the fundamental property to test. It turns out that checking *any* property of Boolean-valued functions can be reduced to performing a dictatorship test, provided the right 'proof' is supplied.

The field of property testing asks a simple problem. Given a collection $\mathcal{C}$ of n-bit Boolean-valued functions, how easy is to determine if an arbitrary boolean functions lies in $\mathcal{C}$ using (not necessarily uniformly) random examples or queries of the function. An **n-query testing algorithm** for $\mathcal{C}$ is an algorithm which randomly chooses elements $x_1, \ldots, x_n \in \{0,1\}^n$, queries $f(x_1), \ldots, f(x_n)$, and decides deterministically whether $f \in \mathcal{C}$. An n-query **local tester** with rejection rate $\lambda > 0$ is a test which always accepts f if $f \in \mathcal{C}$, and if $d(f, \mathcal{C}) > \varepsilon$, then $\mathbf{P}(\text{tester rejects } f) > \lambda\varepsilon$. An alternate statement of this is that if f is accepted with probability greater than $1-\varepsilon$, then $d(f, \mathcal{C}) \leqslant \varepsilon/\lambda$.

Example. *We have already seen a first example of a local tester, which is the BLR algorithm, testing whether an arbitrary Boolean function is linear. It has rejection rate 1, though for large acceptance rates (or for functions close to being linear), the tester acts like it has rejection rate* 3.

Example. *A very similar test to the BLR test checks if a function $f : \mathbf{B}^n \to \mathbf{B}$ is affine (that is, if $f = \pm g$ for some linear function g). The class of affine functions can also be describes as the set of functions f such that for any inputs x, y, z, $f(xyz) = f(x)f(y)f(z)$, for certainly all affine functions satisfy this property, and conversely, if a function satisfies this property, then the function $g(x) = f(x)f(1)$ is linear, because*

$$g(xy) = f(xy)f(1) = f(1xy)f(1) = [f(x)f(1)][f(y)f(1)] = g(x)g(y)$$

Thus, given a function $f : \mathbf{B}^n \to \mathbf{B}$, we can consider a 4-query property test which uniformly picks X, Y, and $Z \in \mathbf{B}^n$, and then tests if $f(XYZ) = f(X)f(Y)f(Z)$. The chance of the test passing f is then

$$\begin{aligned}\mathbf{E}\left(\frac{1 + f(XYZ)f(X)f(Y)f(Z)}{2}\right) &= \frac{1}{2} + \frac{1}{2}\mathbf{E}[f(XYZ)f(X)f(Y)f(Z)] \\ &= \frac{1}{2} + \frac{1}{2}\sum \widehat{f}(S)^4\end{aligned}$$

If the test passes with probability greater than $1 - \varepsilon$, then we conclude

$$\sum \widehat{f}(S)^4 > 1 - 2\varepsilon$$

If $d(f, \mathcal{A}) > \varepsilon$, then $d(f, x^S) > \varepsilon$ and $d(f, -x^S) > \varepsilon$ for each linear function x^S, hence $|\widehat{f}(S)| < 1 - 2\varepsilon$, and so

$$1 - 2\varepsilon < \sum \widehat{f}(S)^4 < (1 - 2\varepsilon)^2 = 1 - 4\varepsilon + 4\varepsilon^2$$

Hence $4\varepsilon^2 - 2\varepsilon = 2\varepsilon(2\varepsilon - 1) > 0$. But this implies $\varepsilon > 1/2$, which is impossible, since any function is $1/2$ close to some affine function. Thus we conclude $d(f, \mathcal{A}) \leqslant \varepsilon$, so the affine checker is a rate one, four query local tester.

We could also consider adaptive testing algorithms, which allow us to choose each x_i sequentially after viewing $f(x_1), \ldots, f(x_{i-1})$, or we could consider local testors with more general rejection rates (quadratic or logarithmic instead of linear, for instance), and to allow the number of queries to depend on the error rate. For now, we'll focus on linear local testers.

4.6 Dictator Tests

Recall that a dictator function is a character x^i, for some $i \in [n]$, and we shall denote the set of dictators by $\mathcal{D}$. We shall find that the **dictator testing** problem, which asks us to determine whether an arbitrary Boolean-valued function is a dictator, is an incredibly versatile property test in the field of computing science.

A dictator test could surely start by testing for linearity. This would then reduce the problem to testing whether an arbitrary character x^S is a dictator. Classically, this was essentially the only method for constructing dictator tests. For instance, aside from the constant functions, the dictators are the only characters which satisfy

$$\mathrm{and}(x^S, y^S) = \prod_{i \in S} \mathrm{and}(x_i, y_i)$$

This requires a complicated analysis, however. We will use our previous results to obtain a more powerful local test.

Using Arrow's theorem, and a bit of extra work, we can come up with a much more intelligent test. Given a Boolean-valued function f, take three random inputs X, Y, and Z uniformly from the 6 possible random triples (X, Y, Z) such that $\mathrm{NAE}(X, Y, Z)$ holds, and compute $\mathrm{NAE}(f(X), f(Y), f(Z))$. Accept f if the outputs are not all equal. We have shown that if f is accepted with probability $1 - \varepsilon$, then $\mathbf{W}^1(f) \geqslant 1 - (9/2)\varepsilon$, and the FKN theorem then implies that f is $O(\varepsilon)$ close to a dictator or its negation. However, this isn't exactly a dictator test, and requires a very difficult theorem to obtain a guarantee on closeness. Fortunately, if we do a NAE test and a BLR test sequentially, then we obtain a 6-query dictator test, and we don't even need to use the FKN test in our analysis to determine the local rate.

Theorem 4.6. *Suppose that $f : \mathbf{B}^n \to \mathbf{B}$ passes both the BLR test and the NAE test with probability $1 - \varepsilon$. If $\varepsilon < 0.1$, then f is ε close to being a dictator. Alternatively, if $d(f, \mathcal{D}) > \varepsilon$, the for $\varepsilon < 0.1$, then f is rejected with probability greater than ε.*

Proof. We then know that f passes the BLR test with at least probability $1 - \varepsilon$, hence there is S such that $d(f, x^S) \leqslant \varepsilon$, and f passes the NAE test with probability $1 - \varepsilon$, hence $\mathbf{W}^1(f) \geqslant 1 - (9/2)\varepsilon$. If S contains k elements, then $\mathbf{W}^k(f) \geqslant \widehat{f}(S)^2 \geqslant (1 - 2\varepsilon)^2 = 1 - 4\varepsilon + 4\varepsilon^2$. If $k \neq 1$, then we must have

$$1 - 4\varepsilon + 4\varepsilon^2 \leqslant (9/2)\varepsilon$$

which implies that $\varepsilon \geqslant 0.1$. If $k = 1$, then $x^j \in \mathcal{D}$, hence $d(f, \mathcal{D}) \leqslant \varepsilon$. $\square$

Thus in general, we find that if f passes the BLR test and NAE test with probability greater than $1 - \varepsilon$, then f is 10ε close to a dictator, for if $\varepsilon < 0.1$, then f is ε close to a dictator, and if $\varepsilon \geqslant 0.1$, then f is $10\varepsilon \geqslant 1$ close to *any function*, hence a dictator of our choice. Thus the BLR and NAE test is a 0.1 rate 6-query local property tester. Note that we aren't that interested in finding local testers with low rejection rate, hence we are allowed to be sloppy with our analysis as above. Instead, we want to show that local testers exist for some property, and use as few queries to the function as possible.

There is a simple trick, given some property test for a class $\mathcal{C}$ with rate λ which employs numerous different subtests $T_1, \ldots, T_m$, is to pick *exactly one* of these tests at random, run the input on this test, and accept if the test passes. By a union bound, we then have

$$\mathbf{P}(\text{new test rejects } f) = (1/m)\sum \mathbf{P}(T_i \text{ rejects } f) \geqslant (1/m)\mathbf{P}(\text{old test rejects } f)$$

Suppose the old test has rejection rate λ. Then if $d(f, \mathcal{C}) > \varepsilon$, then the old test rejects f with probability greater than $\lambda\varepsilon$, and therefore the new test rejects f with probability $(\lambda/m)\varepsilon$. Thus the new test has rejection rate λ/m.

Example. *We have a* 0.1 *rate 6-query dictator test composed of two different tests, each applying 3-queries. Therefore we can use the reduction trick to obtain a* 0.05 *rate 3-query dictator test. Since the 6-query test is really rate* 1 *for* $\varepsilon < 0.1$, *the 3-query test is rate* 0.5 *for* $\varepsilon < 0.1$.

One nice property of dictatorship testing is that we can use a dictatorship test to obtain a 3-query property test over any subproperty $\mathcal{C} \subset \mathcal{D}$. Consider performing the following two tests on an input function f.

1. We perform a 3-query 0.05 rate dictatorship test on f.
2. Form the string y by $y_i = -1$ if $x^i \in \mathcal{C}$, and $y_i = 1$ otherwise. Then $y^j = -1$ if and only if $x^j \in \mathcal{C}$. We perform local decoding on f to calculate the value of f if it was linear.

We can combine the two tests to obtain a 3-query test over $\mathcal{C}$, but it is nicer to calculate the rejection rate assuming we perform both tests, and then

double this rejection rate using the reduction trick. If the test passes with probability at least $1-\varepsilon$, then both the dictator test and the local decoding test pass with probability $1-\varepsilon$. If we let $\delta = 1$, for $\varepsilon < 0.1$, and $\delta = 20$, for $\varepsilon \geqslant 0.1$, we conclude that $d(f, x^j) \leqslant \delta\varepsilon$ for some dictator x^j. If $x^j \in \mathcal{C}$, we are done. Otherwise, if $x^j \notin \mathcal{C}$, then the local decoding calculates y^j from f accurately with probability at least $1 - 2\delta\varepsilon$, and we conclude that the test rejects f with probability at least $1 - 2\delta\varepsilon$. But then

$$1 > (1-\varepsilon) + (1 - 2\delta\varepsilon) = 2 - (1+2\delta)\varepsilon$$

hence $\varepsilon > 1/(1+2\delta)$. Thus if $\varepsilon \leqslant 0.1$, we conclude that f is ε close to a dictator, and so we have a 0.1 rate 6-query dictator test. Using the reduction trick, we obtain a 0.05 rate 3-query dictator test.

4.7 Probabilistically Checkable Proofs

We have shown that any subset of dictator functions has a 3 query proof. It turns out that testing over *any property* can be reduced to dictator testing, provided we have an appropriate 'proof'. To see this, we must slightly generalize our notion of property testing. Rather than just considering families of functions $f : \mathbf{B}^n \to \mathbf{B}$ to test over, we will be testing over binary strings of a fixed length – elements of $\{0,1\}^n$ for a fixed n. Rather than evaluated a function at positions, we get to randomly query certain bits of the string, and then choose to accept or reject the string. The rejection rates, query amount, and so on all generalize, because testing the value of a function at a certain point of the domain is equivalent to looking at a single bit of the binary string.

Example. *Consider testing whether a string is the all zero string. Given an input x, consider randomly selecting an index i, and then testing whether $x_i = 0$. If x is accepted with probability $1-\varepsilon$, then $1-\varepsilon$ percent of the bits in the string x are equal to 0, hence $d(x,0) = \varepsilon$, so this is a test with rate 1, and with a single query.*

Example. *What about testing if the values of a string are all the same. The natural choice is two queries – given x, we pick i and j and test whether $x_i = x_j$.*

If $\min(d(f,0),d(f,1)) = \varepsilon$, *then*

$$\begin{aligned}\mathbf{P}(f \text{ is accepted}) &= \mathbf{P}(x_i = 0 | x_j = 0)\mathbf{P}(x_j = 0) + \mathbf{P}(x_i = 1 | x_j = 1)\mathbf{P}(x_j = 1) \\ &= d(f,0)^2 + d(f,1)^2 \\ &= 1 - 2d(f,0)[1 - d(f,0)]\end{aligned}$$

and so the test is rejected with probability $2\varepsilon(1-\varepsilon)$. *If* $\varepsilon < \alpha$, *then the test is rejected with probability greater than* $2(1-\alpha)\varepsilon$, *so this is a local tester with rate* $\max_{\alpha\in(0,1)} \min(\alpha, 2(1-\alpha)) = 2/3$.

Example. *Consider the property that a string has an odd number of 1s. Then the only local tester for this property must make all n queries in order to be successful.*

Now suppose that, in addition to $x \in \{0,1\}^n$, you are given some $\Pi \in \{0,1\}^m$ which is a 'proof' that x satisfies the required property. Then it might be possible to check both x and Π at the same time, *without ever having to trust the accuracy of* Π, and be able to obtain a higher rate algorithm that x is accepted, assuming that Π is the correct certificate. This is a **probabilistically checkable proof of proximity**, analogous to the PCP verifiers we previously encountered. Specifically, an n-query, length m probabilistically checkable proof of proximity system for a property $\mathcal{C}$ with rejection rate λ is an n-query test, which takes a bit string x, and a proof $\Pi \in \{0,1\}^m$, such that if $x \in \mathcal{C}$, there is a proof Π such that (x,Π) is always accepted, and if $d(x,\mathcal{C}) > \varepsilon$, then for *every* proof Π, (x,Π) is rejected with probability greater than $\lambda\varepsilon$. The converse is that if there is a proof Π such that (x,Π) is accepted with probability greater than $1-\varepsilon$, then $d(x,\mathcal{C}) \leqslant \varepsilon/\lambda$. We normally care about reducing n as much as possible, without regard to m or λ. It is essentially a version of a PCP verifier which only works for languages which form a subset of the class of all Boolean functions, but whose rejection rates are linearly related to the Hamming distance on the set of Boolean functions.

Example. *The property* $\mathcal{O}$ *that a string has an odd number of 1s cannot be checked with few queries without a proof, but we have an easy 3-query PCPP system with proof length* $n-1$, *and rejection rate 1. We expect the proof string to contain the sums* $\Pi_j = \sum_{i\leqslant j+1} x_i$ *in* $\mathbf{F}_2$. *We randomly check that one of the following equation holds*

- $\Pi_1 = x_1 + x_2$.

- $\Pi_j = \Pi_{j-1} + x_{j+1}$.
- $\Pi_{n-1} = 1$.

If $x \in \mathcal{O}$, and Π is a correct proof, then the test is accepted with probablity 1. If $x \notin \mathcal{O}$, then $d(x,\mathcal{O}) = 1/n$, and at least one of these equations fails, hence the probability that x is rejected is at least $1/n$. Thus this is a 3-query rate 1 PCPP tester.

Theorem 4.7. *Any property over length n bit strings has a 3-query PCPP tester with length-2^{2^n} proofs and rejection rate* 0.001.

Proof. Let $\mathcal{C}$ be the required property to test, and consider a bijection $\pi : [N] \to \{0,1\}^n$, where $N = 2^n$. Using π, each element of $\{0,1\}^n$ can be thought of as corresponding to a dictator on $[N]$. Given an input x, we will interpret a proof as a function $\Pi : \{0,1\}^N \to \{0,1\}$ as a dictator corresponding to x. The test then becomes a dictator test, which we can already solve with three queries. Thus we consider two separate tests.

- Testing if Π is a dictator in $\pi^{-1}(\mathcal{C})$.
- Assuming Π is a dictator corresponding to an element $y \in \{0,1\}^n$, testing whether $y = x$.

It is clear that if Π is a correct proof, and $x \in \mathcal{C}$, then the test will always be accepted. Otherwise, if (x,Π) is accepted with probability $1 - \varepsilon$, then the first test tells us that Π is 20ε close to being a dictator in three queries. It remains to find a clever way of determining which element of $\{0,1\}^n$ that the proof Π corresponds to, given that Π is close to some dictator $x^j \in \pi^{-1}(\mathcal{C})$. Local correcting tells us that any calculation we perform over Π will be correct to x_j with probability 40ε. Define $X_i^n = \pi(i)_n \in \{0,1\}^N$. If Π is close to the dictator x^j, then $(X^n)^j = \pi(j)_n$, and if $j = \pi^{-1}(x)$ then $(X^n)^j = x_n$ for all n. Thus our second test will pick n randomly, locally correct Π, and then test if $(X^n)^j = x_n$. If $j \neq \pi^{-1}(x)$, then x^j and $x^{\pi^{-1}(x)}$ agree on half their inputs, and since locally correcting Π to x^j is correct with probability greater than 40ε, our test will reject with probability greater than 80ε.

If Π is a correct proof, then $\Pi(x^i) = \chi_\alpha(x^i)$ TODO finish later. □

The 2^{2^n} is a pretty bad estimate, but it is essentially required to be able to perform this test on all 2^{2^n} different properties on $\{0,1\}^n$. It should be reasonable to be able to identify better systems for 'explicit' properties. This can be qualified by the complexity of the circuit which can compute the indicator function for the property. A **PCPP reduction** is an algorithm which takes a property as input, and outputs a PCPP system testing that property. The notions of proof length, query size, etc all transfer to the PCPP reduction. Essentially, a PCPP reduction gives a constructive proof limiting the ability to test general properties. We have proved that there exists a 3-query 2^{2^n} proof length PCPP reduction with rejection rate 0.001. With a little work, we can improve this reduction to have proof length $2^{\text{poly}(C)}$, where the size of a property C is the size of the circuit representing it. There is a much deeper and more dramatic improvement to this property.

Theorem 4.8 (PCPP)**.** *There is a 3-query PCPP reduction with proof length polynomial in the size of the circuit required to represent the property.*

Though the 2^{2^n} PCPP reduction seems unnecessary after we have proved the PCPP theorem, the 2^{2^n} reduction is integral to the proof of the PCPP theorem, and is therefore important even if you want to understand the proof of the PCPP theorem. It is slightly stronger than the PCP theorem.

4.8 Constraint Satisfaction and Property Testing

Recall that an m arity n contraint **satisfaction problem** over a set Ω is defined by a finite set of predicates $f_1, \ldots, f_n : \Omega^m \to \{0,1\}$. The goal is to find the elements of Ω^m which satisfy the most predicates. This is obviously NP complete, since the problem encompasses the SAT problem, so we instead try and find elements of Ω^m which satisfy the most predicates, or near to the most elements. We assume each f_i is an k-junta, for some positive integer k.

Example. *The 3-SAT problem on m variables is a constraint satisfaction problem, where each constraint is a 3 junta.*

Example. *The MAX-CUT problem on a graph with n nodes and m edges is to find a partition of the graph so that the most possible edges cross the partitions. A partition corresponds to an element of assignment $\{0,1\}^n$, and this partition is evaluated against the not equal constraints for any edge (v,w).*

Example. *Given a system of linear equations in* $\mathbf{F}_2$*, each containing exactly 3 variabels, the MAX-E3-LIN problem asks to find an assignment satisfying as many of the constraints as possible.*

The k-query string testing and constraint satisfaction over k juntas are essentially the same problem. Given an instance of constraint satisfaction over Ω^m with constraints $f_1, \ldots, f_n$, and given a fixed solution $x \in \Omega^m$ (which we can view as a string input to test whether the constraints are satisfied over x), consider the algorithm which randomly choosing $i \in [n]$ and accepts if $f_i(x) = 1$. The the probability the algorithm accepts x is $\text{val}(f, x)$. Conversely, if we have a string testing problem on Ω^m, with some randomized tester using some random bitset (uniformly random), then each selection of random bits corresponds to a constraint where the element of Ω^m is tested for correctness (using only k queries), and the value of the corresponding constraint satisfaction problem to the constraints gives the probability that the string testing algorithm is accepted. We have essentially already seen this correspondence in our discussion of the PCP theorem, where we saw that PCP verifiers (analogous to string testers) corresponed to ρ GAP CSPs (analogous to constraint satisfactions over juntas).

4.9 Hastad's Hardness Theorem

The PCP theorem indicates that unless $P = NP$, it is impossible to approximate 3SAT to arbitrarily close constants. But we can find approximation algorithms for SAT for some constants. For instance, if each clause of MAX-E3SAT contains exactly 3 clauses (we call this subproblem MAX-E3SAT), then each clause is satisfied with probability $7/8$, hence in expectation a random assignment will satisfy $7/8$ of the clauses. There is a way to derandomize this algorithm to obtain a deterministic $7/8$ approximation algorithm. Though the approximation is trivial, Hastad's theorem says this is the best approximation possible.

Theorem 4.9 (Hastad)**.** *For any* $\delta > 0$*, if there is an algorithm that returns a 3SAT assignment satisfying at least* $7/8 + \delta$ *clauses for MAX-E3SAT for all instances of the problem where it is possible to satisfy all clauses, then* $P = NP$*.*

He also proved a similar optimal hardness of approximation for MAX-E3LIN, a result we will concentrate on for the rest of this section.

Theorem 4.10 (Hastad)**.** *If there is $\delta > 0$, and an approximation algorithm for MAX-E3LIN such that if $1-\delta$ of the equations can be satisfied, then the algorithm returns an instance satisfying $1/2+\delta$ of the equations, then* $\mathbf{P} = \mathbf{NP}$.

The idea of this theorem is that we can formalize a dictatorship test with a suitably high rejection rate by a reduction to an instance of MAX-E3LIN, and if we could approximate MAX-E3LIN to a factor greater than $1/2$, then we could solve the constraint satisfaction problem exactly, and we would therefore find $\mathbf{P} = \mathbf{NP}$.

We shall prove a slightly simpler form of this theorem. First, we introduce some formalization. Let $f_1, \ldots, f_n$ be predicates over $\mathbf{B}$. If $0 < \alpha < \beta \leqslant 1$, let $\lambda : [0,1] \to [0,1]$ satisfy $\lambda(\varepsilon) \to 0$ as $\varepsilon \to 0$. Suppose that for each n there is a tester for functions $f : \mathbf{B}^n \to \mathbf{B}$ such that

- If f is a dictator, then the test accepts with probability at least β.
- If $\mathrm{Inf}_i^{1-\varepsilon}(f) \leqslant \varepsilon$, the test accepts with probability at most $\alpha + \lambda(\varepsilon)$.
- The tester can be viewed as an instance of MAX-CSP(f).

Then we call this family an **(α, β) Dictator vs. No Notables Test usings the predicates** f. It is essential that $\lambda = o(1)$, at a rate independent of n, but we care very little about the rate of convergence to zero. There is a Blackbox result which relates Dictator vs. No Notables to hardness of approximation results.

Theorem 4.11. *Fix a CSP over the domain* $\mathbf{B}$ *with predicates $f_1, \ldots, f_n$. Given an (α, β) dictator vs. no notables test using predicates f_i, then for all $\delta > 0$, either the universal game conjecture holds, or there is a $(\alpha + \delta, \beta - \delta)$ approximation for the CSP.*

Given a constraint satisfaction problem f, we define an (α, β) approximation algorithm to be the problem of coming up with an α approximation algorithm for the constraints f, subject to the constraint that we can assume that $\mathrm{val}(f) \geqslant \beta$. We rely on the following theorems, beyond our scope, to prove Hastad's theorem, which says that there is no $(1/2+\delta, 1-\delta)$ approximation algorithm for MAX-E3LIN unless the universal game conjecture holds. Because of the above theorem, it suffices to construct a $(1/2, 1)$ Dictator vs No Notables test using MAX-E3LIN constraints. This is exactly a test for some CSP accepts dictators almost surely, and if the

stable influence $\mathrm{Inf}_i^{1-\varepsilon}(f) \leqslant \varepsilon$, then the test accepts with probability at most $1/2 + o(1)$, independent of n.

We cannot use the BLR test to find a $(1/2, 1-\delta)$ dictator vs no notables test using 3 variable $\mathbf{F}_2$ linear equations, because it doesn't accept functions with no notable coordinates with probability close to $1/2$. The two problems are the constant 0 function and large parity functions. This is fixed by the odd BLR test. Given query access to $f : \mathbf{B}^n \to \mathbf{B}$, we pick $X, Y \in \mathbf{B}^n$ randomly, find $z \in \mathbf{B}$ randomly, and set $Z = XY(z, \ldots, z)$. We then accept the test if $f(X)f(Y)f(Z) = z$. It is easy to show that

Theorem 4.12. *The odd BLR test accepts f with probability*

$$\frac{1}{2} + \frac{1}{2} \sum_{|S| \text{ odd}} \widehat{f}(S)^3 \leqslant \frac{1}{2} + \frac{1}{2} \max_{|S| \text{ odd}} \widehat{f}(S)$$

Thus the odd BLR test rules out constant functions, but still accepts large parity functions. Hastad's idea was to add some noise to z on the left hand side of the equation, while still testing relative to z, so that the stability of high parity functions is knocked off. If f is a dictator, then f has high stability, and there is only a $\delta/2$ chance this will affect the test. On the other hand, if f is x^S, where S is large, there is a high chance this will disturb the chance of passing the test.

In detail, Hastad's δ test takes $X, Y \in \mathbf{B}^n$ and $b \in \mathbf{B}$ uniformly, sets $Z = b(XY)$, choses $Z' \sim N_{1-\delta}(Z)$, and accepts if $f(X)f(Y)f(Z') = (Z, \ldots, Z)$. We claim this is a $(1/2, 1-\delta/2)$ dictator vs. no notables test. We calculate

$$\begin{aligned}
\mathbf{P}(\text{Hastad accepts } f | b = 1) &= \mathbf{E}\left[\frac{1 + f(X)f(Y)f(Z')}{2}\right] \\
&= \frac{1 + \mathbf{E}[f(X)f(Y)T_{1-\delta}(f(XY))]}{2} \\
&= \frac{1 + \mathbf{E}[f(X)(f * T_{1-\delta}(f))(X)]}{2} \\
&= \frac{1}{2} + \frac{1}{2}\sum (1-\delta)^{|S|} \widehat{f}(S)^3
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{P}(\text{Hastad accepts } f | b = -1) &= \mathbf{E}\left[\frac{1 - f(X)f(Y)f(Z')}{2}\right] \\
&= \frac{1}{2} - \frac{1}{2}\sum (-1)^{|S|}(1-\delta)^{|S|} \widehat{f}(S)^3
\end{aligned}$$

Hence the probability that the Hastad test accepts f is

$$\frac{1}{2} + \frac{1}{2} \sum_{|S| \text{ odd}} (1-\delta)^{|S|} \widehat{f}(S)^3$$

If $\text{Inf}_i^{1-\varepsilon}(f) \leqslant \varepsilon$, we conclude that

$$\sum_{i \in S} (1-\varepsilon)^{|S|-1} \widehat{f}(S)^2 \leqslant \varepsilon$$

Hence

$$\begin{aligned}
\frac{1}{2} + \frac{1}{2} \sum_{|S| \text{ odd}} (1-\delta)^{|S|} \widehat{f}(S)^3 &\leqslant \frac{1}{2} + \frac{1}{2} \left(\sum_{|S| \text{ odd}} \widehat{f}(S)^2 \right) \left(\max_{|S| \text{ odd}} (1-\delta)^{|S|} \widehat{f}(S) \right) \\
&= \frac{1}{2} + \frac{1}{2} \sqrt{\max (1-\delta)^{2|S|} \widehat{f}(S)^2} \\
&= \frac{1}{2} + \frac{1}{2} \sqrt{\max (1-\delta)^{|S|-1} \widehat{f}(S)^2} \\
&= \frac{1}{2} + \frac{1}{2} \sqrt{\max_i \text{Inf}_i^{1-\varepsilon}(f)} \\
&= \frac{1}{2} + \frac{1}{2} \sqrt{\varepsilon}
\end{aligned}$$

and $\sqrt{\varepsilon} \to 0$ as $\varepsilon \to 0$.

4.10 Håstad's theorem and Explicit Hardness of Approximation Bounds

If the PCP theorem holds, we can construct a $(O(\log n), O(1))$ verifier for any problem in **NP**. However, it is still of interest to try and find explicit bounds on randomness and bit query numbers, rather than asymptotic bounds. This is not just for academic interest, because bounds on the number of bits used gives explicit hardness of approximation bounds on certain problems. Trying to understand these explicit bounds corresponds to the more advanced PCP theorems which have been proven. It was Håstad who found that we can obtain essentially the same result as the PCP theorem with only three queries.

Theorem 4.13. *For every $\delta > 0$ and every* **NP** *language L, there is a $(\log(n), 3)$ PCP verifier for L, such that for every $x \in L$, there is a proof certificate π such that the PCP verifier accepts with probability greater than or equal to $1 - \delta$, and for every $x \notin L$ and every incorrect proof π, the PCP verifier rejects x with probability greater than or equal to $1/2 - \delta$. What's more, the PCP verifier just takes a proof π of length m, chooses $i, j, k \in [m]$ and $b \in \{0, 1\}$ randomly according to some distribution, and accepts if $\pi_i + \pi_j + \pi_k = b$ (modulo 2).*

It is not superfluous that the PCP verifier uses a linear equation as a PCP verifier. Without further work, the standard PCP theorem only allows us to show that problems are hard to approximate up to some unspecified constant. But Håstad's theorem enables us to get a precise approximation bound for the problem MAX-E3LIN, which takes as input a sequence of linear equations $x_i + y_i + z_i = b_i$, where x_i, y_i, z_i are variables, and $b_i \in \mathbf{F}_2$, and tries to find the maximal subset of equations which can be simultaneously satisfied over $\mathbf{F}_2$. Note that unlike the E3LIN, which asks to decide whether all linear equations can all be simultaneously solved, MAX-E3LIN is NP complete. Håstad shows that there is a known constant such that, if we can approximate MAX-E3LIN past this ratio, the $\mathbf{P} = \mathbf{NP}$.

Theorem 4.14. *If there is a $1/2 + \rho$ approximation algorithm for MAX-E3LIN, for any $\rho > 0$, then $\mathbf{P} = \mathbf{NP}$.*

Proof. The PCP verifier that Håstad constructed implies that any problem L is equivalent to an instance of MAX-E3LIN with $2^{O(\log n)} = \text{poly}(n)$ equations, where either $1 - \delta$ of the equations are solvable, or at most $1/2 - \delta$ of the equations are. If we had a μ approximation algorithm for MAX-E3LIN, where if we can satisfy C of the clauses, then our algorithm returns an assignment satisfying greater than or equal to μC of the clauses. Provided that μC is greater than $1/2 - \delta$ whenever $C \geqslant 1 - \delta$, then we can apply the PCP verifier reduction to solve any **NP** problem in polynomial time. Thus if $\mathbf{P} \neq \mathbf{NP}$ we find $\mu(1 - \delta) < 1/2 - \delta$, so that $\mu < (1/2 - \delta)/(1 - \delta)$, and since δ is arbitrary, we can let $\delta \to 0$ to conclude that $\mu \leqslant 1/2$. □

Håstad's method for proving the inapproximability of MAX-E3LIN has proved to be the main way to prove precise inapproximability results about any algorithm that can be formulated as a CSP problem. First, we show that there is a PCP verifier for every problem in **NP** which takes a proof π which can be viewed as an instance of the CSP problem, and

tests if a single, uniformly randomly chosen constraint holds (normally obtained by taking some **NP** complete problem and forming a PCP verifier for it). This implies a separation bound for reducing problems in **NP**, and if we had a good approximation ratio for the algorithm, this would imply that we could solve these **NP** problems.

The $1/2 + \rho$ inapproximability result for MAX-E3LIN is a tight threshold result, in the sense that we know there is a $1/2$ approximation algorithm for MAX-E3LIN, so the study of approximations for MAX-E3LIN is essentially solved.

Theorem 4.15. *There is a 1/2 approximation algorithm for MAX-E3LIN.*

Proof. TODO □

Using the PCP theorem, we can reduce MAX-E3LIN to MAX-3SAT robustly, in the sense that approximation algorithms to MAX-E3LIN get converted to approximation algorithms to MAX-3SAT. Thus we obtain hard approximation bounds for MAX-3SAT.

Theorem 4.16. *If there is a $7/8 + \rho$ approximation algorithm for MAX-3SAT, for any $\rho > 0$, then $\mathbf{P} = \mathbf{NP}$.*

Proof. Consider determining if more than $1 - \delta$ of the clauses in MAX-E3LIN are satisfied, or if at most $1/2 + \delta$ of the clauses can be satisfied. For $a, b, c \in \mathbf{F}_2$, $a + b + c = 0$ holds if and only if

$$\neg a \vee b \vee c \qquad a \vee \neg b \vee c \qquad a \vee b \vee \neg c \qquad \neg a \vee \neg b \vee \neg c$$

are simultaneously satisfied. Similarily, $a + b + c = 1$ holds if and only if

$$a \vee b \vee c \qquad \neg a \vee \neg b \vee c \qquad \neg a \vee b \vee \neg c \qquad a \vee \neg b \vee \neg c$$

hold simultaneously. Thus if the original E3LIN instance contains n equations, then the constructed instance of 3SAT contains $4n$ clauses, and if m of the E3LIN equations can be satisfied, then at most $4m+3(n-m) = m+3n$ of the clauses of the converted 3SAT problem can be satisfied. Thus deciding whether $1 - \delta$ of the equations in MAX-E3LIN can be satisfied, or whether less than $1/2 + \delta$ of the equations can be satisfied reduces to determining whether more than $(1/2 + \delta)n + 3n$ of the equations of the 3SAT problem can be satisfied, so unless $\mathbf{P} = \mathbf{NP}$, if we have a μ approximation algorithm for 3SAT, then $4\mu n < (1/2 + \delta)n + 3n$, hence $\mu < 7/8 + \delta/4$, and we can let $\delta \to 0$ to conclude $\mu \leqslant 7/8$. □

Note that this theorem only requires the use of 3SAT where each clauses has exactly three variables, so this version of the problem is equally hard. What's more, there are elementary algorithms for 3SAT with exactly three variables which give $7/8$ approximation algorithms, so that this is a tight bound on the approximation.

Theorem 4.17. *There is a $7/8$ approximation algorithm for 3SAT.*

Proof. TODO □

To prove Håstad's theorem, we need to discuss the constraint satisfaction problem on a more general alphabet than the boolean satisfaction we discussed over the PCP theorem. Given a finite alphabet W of size n and a collection of m-juntas functions $f_i : W^k \to \{0,1\}$, we ask if we can find an assignment $x \in W^k$ which maximizes the number of f_i with $f_i(x) = 1$. We let $\text{val}(f)$ denote the maximum fraction of the constraints f which can be satisfied. This is the m-CSP_n problem. As with the proof of the equivalence of the PCP theorem to certain hardness of approximation results for 3SAT, we introduce the YES ρ-GAP m-CSP_n problem as determining whether a given constraint problem f has $\text{val}(f) = 1$, and the NO ρ-GAP m-CSP_n problem as determining whether a given constraint problem has $\text{val}(f) < \rho$, over a fixed language W of size n. The next theorem is a key component of the PCP theorem.

Theorem 4.18. *ρ-GAP 2-CSP_W is hard to approximate for some $0 < \rho < 1$ and some alphabet W.*

We obtain Håstad's result by showing that we can decrease our choice of ρ without increasing the size of W too much, while still maintaining the approximation hardness of the gap. The importance of the CSP in the theorem of Håstad is because of the following reduction, due to Ran Rad, which is a stronger version of the last theorem.

Theorem 4.19. *There is a $c > 1$ such that for every $t > 1$, 2^{-t}-GAP 2-$\text{CSP}_{2^{ct}}$ is* **NP** *hard, and we need only work over instances of 2-CSP_n which are* **regular**, *and have the* **projection property**, *in the sense that every variable is relavant in the same number of constraints, and for any constraint f_i, which can be viewed as a function taking two variables, then for each $x \in \Omega$, there is a unique $y \in \Omega$ such that $f_i(x,y) = 1$. If we form a graph G whose variables are the indices of the problem, and we draw an edge between two indices i and j if they*

are relavant to some common constraint f_k, the projection property implies that there is a bijection $\pi_{ij} : \Omega \to \Omega$ such that $f_k(x, \pi_{ij}(x)) = 1$, and $\pi_{ij}(x)$ is the only value satisfying this constraint. In this case we can forget about the constraints, taking a graph G and assuming that with each edge (i, j) there is a bijection $\pi_{ij} : \Omega \to \Omega$, and we have to find a map $f : G \to \Omega$ which maximizes the number of edges (i, j) such that $f(j) = \pi_{ij}(f(i))$. We can view the map f as a coloring of the graph, and in this formulation, we call the problem the **unique label cover problem**. *The regularity property implies that we may assume our instances of the unique label cover problem are graphs that are regular.*

Thus it suffices to find a $(O(\log n), 3)$ verifier for 2-CSP$_n$ which is sufficiently accurate to maintain the gap constructed in the theorem, so that we can extend this verifier to all **NP** problems. We also rely on a fact that is a key part of the path to proving the full PCP theorem.

Theorem 4.20. *For every two positive integers n and l, there is m and ε_0 such that every n-CSP$_2$ instance f can be reduced to a 2-CSP$_m$ instance g in polynomial time, preserving satisfiability and multiplying the number of constraints by a bounded, constant factor, such that if $val(f) \leqslant 1-\varepsilon$, for $\varepsilon < \varepsilon_0$, then $val(g) \leqslant 1 - l\varepsilon$.*

Thus if ρ-GAP 2-CSP$_m$ is **NP** hard for all m, the above theorem shows that there is ν such that ν-GAP n-CSP$_2$ is **NP** hard, and thus the PCP theorem holds. Håstad provides a verifier for the regular instances of 2-CSP$_n$ using only three queries, showing

The proof of Håstad's theorem, like the PCP lemma we proved earlier, requires a particular application of an error correction code to encode the data in a problem as a proof, in a way which maximizes the amount of data we obtain in a query. The **long code** on $[n]$ encodes each integer $m \in \{1, \ldots, n\}$ as the dictator $x^m : \mathbf{B}^n \to \mathbf{B}$. This is a doubly exponential length encoding, since m is normally written with $\log n$ bits, but is now written with 2^{2^n} bits, but allows us to reduce checking properties of arbitrary $\log n$ bit strings into checking properties of dictators. By property testing, for any $\rho < 1$, given an arbitrary function $f : \mathbf{F}_2^n \to \mathbf{F}_2$, there is a verifier making only three queries to the bits of a function f, such that if f is a dictator, the algorithm accepts with probability $1 - \rho$, and if the test accepts with probability greater than $1/2 + \delta$, then

$$\sum \widehat{f}(S)^3 (1 - 2\rho)^{|S|} \geqslant 2\delta$$

hence f is very similar to a dictator. Indeed, an immediate corollary is that if $k = \log(1/\varepsilon)/2\rho$, there is S of cardinality less than or equal to k with $\widehat{f}(S) \geqslant 2\delta - \varepsilon$.

The key idea between Håstad's reduction is a series of property tests, just like in the weak PCP theorem. The first test is essentially a dictator test, which takes a Boolean-valued function f, uniformly randomly picks inputs X and Y, takes $Z \sim \mathbf{N}_{1-2p}(1)$, and tests if $f(X)f(Y) = f(XYZ)$. If $f = \chi_i$ is a dictator, then $X_i Y_i = X_i Y_i Z_i$ with probability $1 - \rho$. Conversely, we have the following low degree bound if the test passes with high probability, obtained by taking the required Fourier expansion.

Theorem 4.21. *If the test passes on f with probability $1/2 + \delta$, then $\sum(1 - 2\rho)^{|S|}\widehat{f}(S)^3 \geqslant 2\delta$.*

An instant result is an existence result that if f passes with hihg probability, then it is close to some dictator.

Corollary 4.22. *If f passes the long code test with probability $1/2 + \delta$, then for $k = \log(1/\varepsilon)/2\rho$, there is S with $|S| \leqslant k$, and $\widehat{f}(S) \geqslant 2\delta - \varepsilon$.*

Håstad reduction takes a particular NP complete problem, and provides a verifier for it. Thanks to Rad's theorem, we can assume this NP complete problem is an instance of a regular label cover problem with graph G with edge constraints π_{ij}, where we may further assume that either $\mathrm{val}(G) = 1$, or $\mathrm{val}(G) < \varepsilon$ for an arbitrarily small ε, and the alphabet is some set $[n]$. We expect a proof Π for the label cover problem to be a satisfying assignment, where we encode each assignment to each variable as the long code, where we assume the long codes we obtain are odd (which can always be done if we cut the encoding in half). The basic test of correctness picks two vertices v and w, obtains the long codes f and g corresponding to v and w, and then tests if f encodes x and g encodes y, where $\pi_{vw}(x) = y$.

If the instance of Label cover is completely satisfiable, and f and we know the algorithm accepts with probability greater than or equal to $1 - \rho$ (this is just the test we considered above). TODO: FINISH.

4.11 MAX-CUT Approximation Bounds and the Majority is Stablest Conjecture

Recall the MAX-CUT problem, which asks, given a graph G, to find a partition of the vertices of G into two sets V and W, such that the cardinality of the cut

$$\delta(V_0, V_1) = \{(v, w) \in E : v \in V, w \in W\}$$

is maximized. The problem is known to be NP-complete, and here we will derived tight approximation bounds for the problem. Using the techniques of semidefinite programming, one can derive an α-approximation algorithm, where

$$\alpha = \min_{0<t<\pi} \frac{2t}{\pi(1-\cos t)} \approx 0.878567$$

It may be surprising that geometry becomes involved in the MAX-CUT problem, but we will see the impact of geometry in our analysis that α is a tight constant approximation bound for the problem, assuming that the 'unique games conjecture' holds.

The unique games conjecture can be stated in terms of the inapproximability of a particular **NP** complete problem, known as the unique label cover problem. We are given a bipartite graph G with left vertices V and right vertices W, an alphabet Σ, and for each $(v, w) \in E$, a bijection $\pi_{(v,w)} : \Sigma \to \Sigma$. A labelling $L : (V \cup W) \to \Sigma$ then 'satisfies' an edge (v, w) if $\pi_{(v,w)}(L(v)) = L(w)$, and the goal of the unique label cover problem is to find a label which satisfies as many edges as possible. This problem can be formalized as a version of MAX-CSP, where each constraint, indexed by $(v, w) \in E$, is

$$f_{(v,w)}(L) = "\pi_{(v,w)}(L(v)) = L(w)"$$

which is

or alternatively, as a way to reduce any problem to a 2-query verifier

NP hardness result for

In order to obtain tight approximation bounds on the MAX-CUT problem, we introduce the techniques involves with the unique games conjecture, and its relation to the majority is stablest theorem.

4.12 Max Cut Inapproximability

Recall that the MAX-CUT problem asks, given a graph G, to find the partition V, W of the vertices of the graph which maximizes the cardinality of edges crossing the cut, which is defined to be

$$\delta(V, W) = \{(v, w) \in E : v \in V, w \in W\}$$

As with many **NP**-complete optimization problems, MAX-CUT can be modelled as an instance of MAX-CSP over the alphabet $\{0, 1\}$, where the constraints are $v \neq w$, where $(v, w) \in E$. Semidefinite programming gives an α approximation algorithm for MAX-CUT, where

$$\alpha = \min_{0<x<\pi} \frac{2x}{\pi(1 - \arccos x)}$$

We shall show that this is a tight bound for any approximation algorithm to MAX-CUT. As we have seen, the standard way to obtain an approximation bound for any k-CSP is to define a k query PCP verifier for some **NP** complete language L over an alphabet Σ^* obtained by considering a single predicate for MAX-CUT. If the PCP verifier accepts an element of L with probability greater than or equal to $1 - \varepsilon$, and the PCP verifier rejects any other string with probability greater than $1 - \delta$, then we obtain that MAX-CUT cannot have an α approximation algorithm for $\alpha > \delta/(1 - \varepsilon)$ unless $\mathbf{P} = \mathbf{NP}$. However, our current knowledge of standard techniques for choosing the language L breaks down in the case where $k = 2$. One of the most important conjectures in complexity theory, second only to the $\mathbf{P} = \mathbf{NP}$ conjecture, is that there is a language L which is very applicable to 2-CSPs.

The problem UNIQUE-LABEL-COVER over an alphabet Σ is a 2-CSP defined with respect to a bipartite graph G with vertices $V \cup W$, such that for each edge (v, w) in the graph, we have a bijection $\pi_{(v,w)} : \Sigma \to \Sigma$, and the constraints are of the form $v = \pi_{(v,w)}(w)$, for each edge (v, w), where the variables are the vertices $V \cup W$ of the graph G. It is incredibly important result to most work in complexity theory that UNIQUE-LABEL-COVER is hard to approximate.

Theorem 4.23 (The Unique-Games Conjecture)**.** *For any $\eta, \gamma > 0$, there is an alphabet Σ such that it is* **NP** *hard to determine if an instance X of*

UNIQUE-LABEL-COVER over Σ has $val(X) \geqslant 1-\eta$, or $val(X) \leqslant \gamma$. That is, for any problem L in **NP** *over strings in Π^*, there is a polynomial time computable function f taking strings in Π^* to instances of UNIQUE-LABEL-COVER over Σ, such that if $s \in L$ then $val(f(s)) \geqslant 1-\eta$, and if $s \notin L$ then $val(f(s)) \leqslant \gamma$.*

In this formulation, the unique game conjecture has a 2-query PCP verifier, such that each proof of an input x is encoded as an assignment to the 2-CSP $f(x)$, and we simply check if the assignment satisfies $v = \pi_{(v,w)}(w)$ for some random edge (v,w). Since f is polynomial time computable, we need only $O(\log n)$ random bits for this verifier.

The second 'conjecture' we will require was actually proved only in 2005, and is a statement about the properties of 'balanced' boolean functions.

Theorem 4.24 (Majority is Stablest)**.** *If $\rho \in [0,1)$, then for any $\varepsilon > 0$, there is $\delta > 0$ such that if $f : \mathbf{B}^n \to [-1,1]$ is an unbiased function, and $Inf_i(f) \leqslant \delta$ for all i, then*

$$Stab_\rho(f) \leqslant 1 - \frac{2}{\pi}\arccos\rho + \varepsilon$$

hence among the balanced functions whose coordinates individually have small influence bounded by δ, we find that

$$Stab_\rho(f) \leqslant \lim_{\substack{n\to\infty \\ n\ odd}} Stab_\rho(Maj_n) + o_\delta(1)$$

hence the majority functions 'maximize' the stability of low influence functions, up to a $o_\varepsilon(1)$ factor.

Theorem 4.25. *The majority is stablest theorem continues to hold if we replace the assumption $Inf_i(f) \leqslant \delta$ with*

$$Inf_i^{\leqslant k}(f) = \sum_{\substack{i\in S \\ |S|\leqslant k}} \widehat{f}(S)^2 \leqslant \delta'$$

where k and δ' are universal functions of ε and ρ.

Proof. Fix $\rho < 1$ and $\varepsilon > 0$. If γ is chosen such that $\rho^k(1-(1-\gamma)^{2k}) < \varepsilon/4$ for all k, and δ is chosen such that if $\mathrm{Inf}_i(g) \leqslant \delta$ then $\mathrm{Stab}_\rho(g) \leqslant 1 -$

$(2/\pi)\arccos\rho + \varepsilon/4$, then let $\delta' = \delta/2$ and choose k' such that $(1-\gamma)^{2k'} < \delta$. If f satisfies $\mathrm{Inf}_i^{\leqslant k'}(f) \leqslant \delta'$ and

Given $0 < \gamma < 1$, if we let $g = T_{1-\gamma}f$, and if we assume that for all i, $\mathrm{Inf}_i^{\leqslant k'}(f) \leqslant \delta'$, for some fixed δ', then we find

$$\mathrm{Inf}_i(g) \leqslant \mathrm{Inf}_i^{\leqslant k'}(f) + \sum_{\substack{i \in S \\ |S| > k'}} (1-\gamma)^{2|S|}\widehat{f}(S)^2 \leqslant \delta' + (1-\gamma)^{2k'}$$

If we choose γ small enough (depending only on δ, our choice of δ', and our choice of k'), then we find that $\mathrm{Inf}_i(g) \leqslant \delta$, and if we choose γ small enough such that $\rho^{|S|}[1-(1-\gamma)]^{2|S|} \leqslant \varepsilon$ for all $|S|$, we find

$$\mathrm{Stab}_\rho(f) = \mathrm{Stab}_\rho(g) + \sum \rho^{|S|}[1-(1-\gamma)]^{2|S|}\widehat{f}(S)^2 \leqslant \mathrm{Stab}_\rho(g) + \varepsilon$$

applying majority is stablest to $\mathrm{Stab}_\rho(g)$ gives the result for f. □

Theorem 4.26. *Majority is stablest implies that for any $\rho \in (-1, 0]$ and $\varepsilon > 0$, there is $\delta > 0$ such that for any boolean function $f : \mathbf{B}^n \to [-1, 1]$, if $Inf_i(f) \leqslant \delta$ for all i, then*

$$Stab_\rho(f) \geqslant 1 - (2/\pi)\arccos\rho - \varepsilon$$

Proof. Given f, let g be the odd part of f, so $g(x) = [f(x) - f(-x)]/2$. Then $\mathbf{E}[g(x)] = 0$, $\mathrm{Inf}_i(g) \leqslant \mathrm{Inf}_i(f)$, and so we may apply majority is stablest for $-\rho$ to conclude

$$\begin{aligned}\mathrm{Stab}_\rho(f) &\geqslant \mathrm{Stab}_\rho(g) = -\mathrm{Stab}_{-\rho}(g) \\ &\geqslant -\left(1 - \frac{2}{\pi}\arccos(-\rho) + \varepsilon\right) \\ &= 1 - \frac{2}{\pi}\arccos(\rho) - \varepsilon\end{aligned}$$

hence the majority is stablest theorem holds for $\rho < 0$. □

Putting both these theorems together, we conclude that

Theorem 4.27. *For any $\rho \in (-1, 0]$ and $\varepsilon > 0$, there is $\delta > 0$ and k such that for any boolean function $f : \mathbf{B}^n \to [-1, 1]$ with $Inf_i^{\leqslant k}(f) \leqslant \delta$ for all i, then*

$$Stab_\rho(f) \geqslant 1 - (2/\pi)\arccos\rho - \varepsilon$$

which is obtained by combining the two theorems above.

Now we construct a 2-query PCP verifier for UNIQUE-LABEL-COVER using MAX-CUT. By a refinement to the unique games conjecture, we can assume that in the instance of UNIQUE-LABEL-COVER is given over a bipartite graph G whose left vertices are regular, in the sense that they all have the same degree, so that choosing a random left vertex v, and then a random neighbour w corresponds to the uniform distribution on the set of edges in G. We assume the instance X of UNIQUE-LABEL-COVER has either $\operatorname{val}(X) \geqslant 1-\eta$, or $\operatorname{val}(X) \leqslant \gamma$. The verifier expects the proof certificate Π to contain the long code $\Pi_w : \mathbf{B}^{\Sigma} \to \mathbf{B}$ for each right vertex w.

If Π_w and $\Pi_{w'}$ are actually the long codes of the labelling for w and w', where the labelling has value greater than or equal to $1-\eta$, then by a union bound we conclude that the constraints corresponding to (v,w) and (v,w') are both satisfied with probability greater than or equal to $1-2\eta$, and conditioning that this is true, the probability that the two are unequal is exactly the probability that $Y=-1$, which occur with probability $(1-\rho)/2$, hence the test accepts with probability $(1-2\eta)(1-\rho)/2$. Conversely, the probability of accepting can be calculated to be exactly

$$\begin{aligned}
&\mathbf{P}(\Pi_w(X\circ\sigma) \neq \Pi_{w'}((X\circ\sigma')Y)) \\
&\quad = \mathbf{E}\left(\frac{1-\Pi_w(X\circ\sigma)\Pi_{w'}((X\circ\sigma')Y)}{2}\right) \\
&\quad = \frac{1}{2} - \frac{1}{2}\mathbf{E}\left(\mathbf{E}[\Pi_w(X\circ\sigma)|X]\mathbf{E}[\Pi_{w'}((X\circ\sigma')Y)|X,v]\right) \\
&\quad = \frac{1}{2} - \frac{1}{2}\mathbf{E}\left[\operatorname{Stab}_\rho\left(\mathbf{E}[\Pi_w(X\circ\sigma)|X,v]\right)\right]
\end{aligned}$$

Denote $\mathbf{E}[\Pi_w(X\circ\sigma)|X,v] = g_v(X)$. It follows that if the probability of success is greater than or equal to $(\arccos\rho)/\pi + \varepsilon$, then by applying Markov's inequality, for at least $\varepsilon/2$ of the left vertices v,

$$\operatorname{Stab}_\rho(g_v) \leqslant 1 - (2/\pi)\arccos\rho - \varepsilon$$

and for each such vertex v (which we call call 'good' vertices), the majority is stablest theorem can be applied to determine that there is some index $s \in \Sigma$ such that $\delta \leqslant \operatorname{Inf}_s^{\geqslant k}(\Pi_v)$, and so

$$\delta \leqslant \sum_{\substack{s\in S\\ |S|\leqslant k}} \widehat{g_v}(S)^2 = \sum_{\substack{s\in S\\ |S|\leqslant k}} \mathbf{E}_w\left[\widehat{\Pi_w}(\sigma^{-1}(S))^2\right] = \mathbf{E}_w\left[\operatorname{Inf}^{\leqslant k}_{\sigma^{-1}(s)}(\Pi_w)\right]$$

For each $w \in W$, let the set C_w of candidate labels by all $s \in \Sigma$ such that $\mathrm{Inf}_s^{\leqslant k}(\Pi_w) \geqslant \delta/2$. Then $|C_w| \leqslant 2k/\delta$, and for each good vertex, v, at least $\delta/2$ of the neighbours w of v have $\mathrm{Inf}_{\sigma^{-1}(s)}^{\leqslant k}(f) \geqslant \delta/2$, and so $\sigma^{-1}(s) \in C_w$. Now we consider a labelling of each vertex $w \in W$ by choosing an element of C_w, if possible, or any label if this set is empty. If follows that on the set of vertices adjacent to good vertices, at least $(\delta/2)(\delta/2k)$ are satisfied in expectation, and therefore there is a labelling which satisfies $(\varepsilon/2)(\delta/2)(\delta/2k)$ of the constraints. If $(\varepsilon/2)(\delta/2)(\delta/2k) > \gamma$, we conclude that the probability that the test accepts is less than $(\arccos\rho)/\pi + \varepsilon$.

The PCP verifier above shows that MAX-CUT is α hard to approximate, where

$$\alpha = \frac{2\arccos\rho + 2\varepsilon}{\pi(1-\rho)}$$

for any $\rho \in (-1, 0)$ and $\varepsilon > 0$. Maximizing over this set, letting $\varepsilon \to 0$ first and then optimizing, we conclude that we may let α be the hardness factor we first considered.

4.13 FOAIJDOIWJ

The standard way to obtain an approximation bound for is to consider a PCP verifier for some **NP**-complete language L over some alphabet Σ^* using the predicates in MAX-CUT. Such a verifier leads to a polynomial time reducible function f taking strings in Σ^* to instances of MAX-CUT, in such a way that if $x \in L$, then $\mathrm{val}(f) \geqslant 1-\varepsilon$, and if $x \notin L$ then $\mathrm{val}(f) \leqslant \delta$. If we had some α approximation algorithm for MAX-CUT, then unless $\mathbf{P} = \mathbf{NP}$, we would have to have $\alpha(1-\varepsilon) \leqslant \delta$, so $\alpha \leqslant \delta/(1-\varepsilon)$. However, it turns out that it is difficult

Recall that if J is an index set, then the long code encodes elements of J to Boolean valued function on $\mathbf{B}^J$, mapping $j \in J$ to the dictator function $x^j : \mathbf{B}^J \to \mathbf{B}$. Given $f : \mathbf{B}^J \to \mathbf{B}$, choose some $X \in \mathbf{B}^J$ at random, and let $Y \sim N_\rho(X)$. Our test will check whether $f(X) \neq f(Y)$. It is easy to arithmetize

$$\mathbf{P}(f(X) \neq f(Y)) = \mathbf{E}\left(\frac{1 - f(X)f(Y)}{2}\right) = \frac{1}{2} - \frac{1}{2}\mathrm{Stab}_\rho(f)$$

If f is a dictator, then $\mathbf{P}(f(X) \neq f(Y)) = (1-\rho)/2$. If f is not a dictator, then

$$\mathrm{Stab}_\rho(f) \geqslant 1 - \frac{2}{\pi}\arccos\rho$$

hence $\mathbf{P}(f(X) \neq f(Y)) \leqslant (1/\pi) \arccos \rho$.

Chapter 5

Hypercontractivity

The goal of hypercontractivity is to obtain precise estimates of the effects of the noise operators T_ρ on the space of Boolean functions. If we can control the noise operator, we can also obtain powerful properties about the noise stability of functions, and this is where the most interesting applications of hypercontractivity occur, such as in the majority is stablest theorem. We can also use this properties to understand extremal problems in the geometry of the Hamming cube, and other discrete spaces.

Given a real-valued random variable X, we say X is M-**reasonable** if $\mathbf{E}[X^4] \leqslant M\mathbf{E}[X^2]^2$, or equivalently, if $\|X\|_4^4 \leqslant M\|X\|_2^4$. This is a scale invariant quantity, since X is M-reasonable if and only if λX is M-reasonable for any $\lambda \neq 0$, so that reasonability measures the *relative* spread of a random variable. We always have $M \geqslant 1$, because using Hölder's inequality, we find $\mathbf{E}[X^2]^2 \leqslant \mathbf{E}[X^4]$.

Example. *If X is uniformly random over $\mathbf{B}$, then $\mathbf{E}[X^2] = \mathbf{E}[X^4] = 1$, so X is 1-reasonable. If X is normally distributed with mean zero and variance one, then $\mathbf{E}[X^2] = 1$, $\mathbf{E}[X^4] = 3$, so that X is 3-reasonable. If X is uniform over $[-1,1]$, then $f_{X^2}(x) = 1/2x^{1/2}$ and $f_{X^4}(x) = 1/4x^{3/4}$, hence $\mathbf{E}[X^2] = 1/3$, and $\mathbf{E}[X^4] = 1/5$, and we find X is $9/5$ reasonable. If X is a highly biased Bernoulli random variable, with $\mathbf{P}(X = 1) = 1/2^n$ and $\mathbf{P}(X = 0) = 1 - 1/2^n$, then $\mathbf{E}[X^2] = \mathbf{E}[X^4] = 1/2^n$, hence X is 2^n-reasonable.*

We can use the inequality which defines M-reasonable random variables to obtain a tail bound sharper than the standard Chebyshev inequal-

ity, using the Markov inequality to find that if $C = \|X\|_2$, then

$$\mathbf{P}(|X| \geqslant Ct) = \mathbf{P}(X^4 \geqslant t^4 C^4) \leqslant \frac{1}{t^4} \frac{\mathbf{E}[X^4]}{\mathbf{E}[X^2]} \leqslant \frac{M}{t^4}$$

More interestingly, we can upper bound the probability that M-reasonable random variables are close to zero. Applying the Paley-Zygmund inequality, we find

$$\mathbf{P}(|X| \geqslant Ct) = \mathbf{P}(X^2 \geqslant C^2 t^2) \geqslant (1 - t^2)^2 \frac{\mathbf{E}[X^2]^2}{\mathbf{E}[X^4]} \geqslant \frac{(1 - t^2)^2}{M}$$

Thus reasonable random variables are neither too concentrated nor too spread out, bounded by a t^4 term. For discrete random variables, this takes the form that the probability distribution is evenly spread out.

Theorem 5.1. *If X is a discrete random variable with probability mass function f_X, and $\mu = \min f_X$, then X is μ^{-1} reasonable.*

Proof. Let $M = \|X\|_\infty$, and. Then

$$\mathbf{E}[X^2] \geqslant \mu M^2 \qquad \mathbf{E}[X^4] \leqslant M^2 \mathbf{E}[X^2]$$

so $\mathbf{E}[X^4]/\mathbf{E}[X^2]^2 \leqslant M^2/\mathbf{E}[X^2] \leqslant 1/\mu$. □

Don't think that the converse to this statement is true, for if $X_n = \sum_{k=1}^n x_k/\sqrt{n}$, where the x_k are independent and uniform on $\mathbf{B}$, and the X_n converge in distribution to a normally distributed random variable with mean zero and variance one, and so we find that the X_n are $3 + o_n(1)$ reasonable. Now given a series of independent random bits, it is an interesting question how to construct a distribution which is unreasonable. The theorem aboves says that we must use many random bits, and since the 2^n unreasonable function we constructed in the example requires a degree n polynomial in the bits, we should expect that low degree polynomials in the bits *are reasonable*.

Lemma 5.2 (Bonami)**.** *For each k, if $f : \mathbf{B}^n \to \mathbf{R}$ has degree $\leqslant k$, and $X_1, \ldots, X_n$ are independent and uniformly random ± 1 bits, then $f(X_1, \ldots, X_n)$ is 9^k reasonable.*

Proof. We assume $k \geqslant 1$, for otherwise f is a constant function, and the theorem is trivial. Write $f(x) = x_n D_n f(x) + E_n f(x) = x_n g(x) + h(x)$, where $\deg g(x) \leqslant k-1$ and $\deg h(x) \leqslant k$. Then

$$\begin{aligned}\mathbf{E}[f(X)^4] &= \mathbf{E}[(X_n g(X) + h(X))^4] \\ &= \mathbf{E}[g(X)^4 + 6g(X)^2 h(X)^2 + h(X)^4 + 4g(X)h(X)X_n(g(X)2 + h(X)^2)] \\ &= \mathbf{E}[g(X)^4] + 6\mathbf{E}[g(X)^2 h(X)^2] + \mathbf{E}[h(X)^4]\end{aligned}$$

Similarily, we find $\mathbf{E}[f(X)^2] = \mathbf{E}[g(X)^2] + \mathbf{E}[h(X)^2]$. By induction, $\mathbf{E}[g(X)^4] \leqslant 9^{k-1}\mathbf{E}[g(X)^2]^2$, and $\mathbf{E}[h(X)^4] \leqslant 9^k \mathbf{E}[h(X)^2]^2$. Applying Cauchy Schwarz, we find

$$\mathbf{E}[g(X)^2 h(X)^2] \leqslant \sqrt{\mathbf{E}[g(X)^4]}\sqrt{\mathbf{E}[h(X)^4]} \leqslant (9^k/3)\mathbf{E}[g(X)^2]\mathbf{E}[h(X)^2]$$

Combing these results, we find

$$\begin{aligned}\mathbf{E}[f(X)^4] &\leqslant 9^k \left(\mathbf{E}[g(X)^2]^2 + 2\mathbf{E}[g(X)^2]\mathbf{E}[h(X)^2] + \mathbf{E}[h(X)^2]^2\right) \\ &= 9^k(\mathbf{E}[g(X)^2]^2 + \mathbf{E}[h(X)^2]^2)^2 = 9^k \mathbf{E}[f(X)^2]^2\end{aligned}$$

and this completes the proof by induction. □

Corollary 5.3. *If $X_1, \ldots, X_n$ are independent, but not necessarily identically distributed M-reasonable random variables, with $\mathbf{E}[X_i] = \mathbf{E}[X_i^3] = 0$, then $f(X_1, \ldots, X_n)$ is $\max(M, 9)^k$ reasonable, where f is a degree k multilinear polynomial.*

The Bonami lemma shows that low degree polynomials in random bits are very reasonable, which allows to easily obtain concentration and anticoncentration bounds. Indeed, if $f : \mathbf{B}^n \to \mathbf{R}$ has degree at most k, has mean μ and standard deviation σ, then

$$\mathbf{P}(|f(X) - \mu| > \sigma/2) \geqslant 9^{1-k}/16$$

Indeed, the function $(f-\mu)/\sigma$ is a degree k polynomial, hence $(f-\mu)/\sigma$ is 9^k reasonable, and the theorem follows by applying the anticoncentration bound for reasonable random variables. This immediately leads to a proof of the FKN theorem we saw in our analysis of the theory of social choice.

Theorem 5.4 (Friedgut-Kalai-Naor)**.** *If $f : \mathbf{B} \to \mathbf{B}$ has $\mathbf{W}^1(f) = 1 - \delta$, then f is $O(\delta)$ close to a dictator or its negation.*

Proof. If $f(x) = \sum a_S x^S$, let $g(x) = \sum a_i x_i$. Then $\mathbf{E}[g(X)^2] = 1 - \delta$. It suffices to show that $\mathbf{V}(g(X)^2)$ is small, because

$$\mathbf{V}(g(X)^2) = \mathbf{E}\left[\left(g(X)^2 - \sum a_i^2\right)^2\right] = \sum_{i \neq j} a_i^2 a_j^2 = \left(\sum a_i^2\right)^2 - \sum a_i^4$$
$$= (1-\delta)^2 - \sum a_i^4 \geqslant (1 - 2\delta) - \sum a_i^4$$

so if $\mathbf{V}(g(X)^2) \leqslant O(\delta)$, then

$$(1 - 2\delta) - O(\delta) \leqslant \sum a_i^4 \leqslant \left(\max a_i^2\right) \sum a_i^2 \leqslant \max a_i^2 \leqslant \max |a_i|$$

hence we conclude that f is $O(\delta)$ close to some dictator or its negation. Now we apply the Bonami inequality, since g is degree 2, to conclude that

$$\mathbf{P}\left(|g(X)^2 - (1-\delta)| \geqslant (1/2)\sqrt{\mathbf{V}(g(X)^2)}\right) \geqslant 1/144$$

If $\mathbf{V}(g(X)^2) \geqslant 4K\delta$ for some large K, then the inequality above says that $|g(X)|$ is frequently $\sqrt{K\delta}$ far away from 1, and since f is $0-1$ valued, we conclude that $|f - g|$ is frequently large. Yet if $|g(X)^2 - 1| > \sqrt{K\delta}$, then $(f(X) - g(X))^2 \geqslant K'\delta$ (for some particular choice of K'), hence $\mathbf{E}[(f(X) - g(X))^2] \geqslant (K'\delta)/144 > \delta$, yet $\mathbf{E}((f(X) - g(X))^2) = \delta$, hence $K' > 144$. □

5.1 Sensitivity of Small Hypercube Subsets

An immediate consequence of the Bonami Lemma is that for any Boolean function $f : \mathbf{B}^n \to \mathbf{R}$,

$$\|T_{1/\sqrt{3}}(f^{=k})\| = \frac{1}{3^{k/2}}\|f^{=k}\|_4 \leqslant \|f^{=k}\|_2$$

this is a special case of a more general theorem.

Theorem 5.5 ((2,4) Hypercontractivity Theorem)**.** *For any Boolean function $f : \mathbf{B}^n \to \mathbf{R}$, then $\|T_{1/\sqrt{3}} f\|_4 \leqslant \|f\|_2$.*

Proof. We simply repeat the induction technique used for the Bonami lemma. TODO □

Thus not only is $T_{1/\sqrt{3}}$ a *contraction* on $L^2(\{-1,1\}^n)$, but also a contraction from L^2 to L^4. This shows that the noise operator preserves reasonable functions in some sense. Though $\|T_{1/\sqrt{3}}f\|_4$ does not appear to have a combinatorial meaning, the two norm is

$$\|T_{1/\sqrt{3}}f\|_2^2 = \langle T_{1/\sqrt{3}}f, T_{1/\sqrt{3}}f\rangle = \langle f, T_{1/3}f\rangle = \mathrm{Stab}_{1/3}(f)$$

since $T_{1/\sqrt{3}}$ is self-adjoint. And the (2,4) hypercontractivity theorem is essentially equivalent to a $(4/3, 2)$ hypercontractivity theorem.

Theorem 5.6 ($(4/3,2)$ hypercontractivity)**.** *For any Boolean function f, $\|T_{1/\sqrt{3}}f\|_2 \leqslant \|f\|_{4/3}$, which means that $Stab_{1/3}(f) \leqslant \|f\|_{4/3}^2$.*

Proof. Applying the (2,4) hypercontractivity theorem, and Hölder's theorem, we find

$$\|T_{1/\sqrt{3}}f\|_2^2 = \langle f, T_{1/3}f\rangle \leqslant \|f\|_{4/3}\|T_{1/3}f\|_4 \leqslant \|f\|_{4/3}\|T_{1/3}f\|_2$$

and the inequality is obtained by dividing by $\|T_{1/\sqrt{3}}f\|_2$. □

If f has range $\mathbf{B}$, the result isn't really that interesting, but if look at functions f with range $\{0,1\}$, which can be seen as identifying a subset of the cube, then we get an interesting corollary.

Corollary 5.7. *If $A \subset \mathbf{B}^n$ has volume ν, and f is the characteristic function of A, then $Stab_{1/3}(f) = \mathbf{P}(X \in A, Y \in A)$, where X is uniformly distributed, and $Y \sim N_{1/3}(X)$. The (4/3,2) hypercontractivity result implies that $Stab_{1/3}(f) \leqslant \nu^{3/2}$, and then $\mathbf{P}(Y \in A) \leqslant \nu^{1/2}$.*

Example. *If $\nu = 2^{-k}$, and A is a subcube of the Hamming cube of codimension k (without loss of generality, the subset consisting of points whose first k coordinates are 1), then for every $x \in A$, when $X \sim N_{1/3}(x)$,then $X \in A$ if and only if the first k coordinates of x don't change, which happens with probability $(2/3)^k = (2/3)^{\log(1/\nu)} = \nu^{\log(3/2)}$.*

Example. *For any positive integer n and $\rho \in [-1,1]$, the* **ρ-stable hypercube graph** *is the complete directed graph on the Hamming cube with edge weights given by assigning the edge (x,y) with the weight $\mathbf{P}((X,Y) = (x,y))$, where (X,Y) is ρ-correlated. If $\rho = 1 - 2\delta$ we also call this graph the δ-noise hypercube graph. We have shown that if we choose a random vertex $x \in A$, and then take a random edge out of x, we end up outside A with probability at least $1 - \sqrt{\nu}$.*

If $g : \mathbf{B}^n \to \{-1, 0, 1\}$ satisfies $\mathbf{P}(g \neq 0) = \mathbf{E}[|g|] = \mathbf{E}[g^2] = \alpha$, then we find $\mathrm{Stab}_{1/3}(g) \leqslant \alpha^{3/2}$, by essentially the same proof as for the characteristic function. Since

$$\mathrm{Stab}_{1/3}(g) = \mathrm{Inf}_i^{1/3}(f)$$

we find that for a Boolean-valued function f, the hypercontractivity result states that $\mathrm{Inf}_i^{1/3} \leqslant \mathrm{Inf}_i(f)^{3/2}$.

Since noise stability measures how 'low' the Fourier weight of a function is, this corollary implies that a Boolean function taking values in $\{0, 1\}$ with mean α cannot have much of its Fourier weight at a low degree. That is, if $\alpha^{3/2} \geqslant \mathrm{Stab}_{1/3}(f) \geqslant (1/3)^k \mathbf{W}^{\leqslant k}(f)$, then $\mathbf{W}^{\leqslant k}(f) \leqslant 3^k \alpha^{3/2}$.

5.2 $(2, q)$ and $(p, 2)$ Hypercontractivity For Bits

We now try to generalize the proof of the (2,4) hypercontractivity theorem to other norms. The reason why hypercontractivity is more easy to generalize than the Bonami lemma is that it doesn't depend on reasonability, which is translation sensitive. For $1 \leqslant p \leqslant q \leqslant \infty$, and $0 \leqslant \rho < 1$, we say that a random variable X is (p, q, ρ)-**hypercontractive** if $\|a + \rho b X\|_q \leqslant \|a + bX\|_p$ for all $a, b \in \mathbf{R}$. By homogeneity, it suffices to check the condition for $a = 1$, and $b \in \mathbf{R}$, or $b = 1$ and $a \in \mathbf{R}$. Furthermore, if X is (p, q, ρ) hypercontractive, it is also (p, q, ν) hypercontractive for $\nu < \rho$.

Theorem 5.8. *If X and Y are independent (p, q, ρ) hypercontractive random variables, then $X + Y$ is also (p, q, ρ) hypercontractive.*

Theorem 5.9. *If X is a random variable uniform over $\mathbf{B}$, then X is $(2, 6, 1/\sqrt{5})$ hypercontractive.*

Proof. We need to show that for $\rho = 1/\sqrt{5}$, then $\mathbf{E}[(a + \rho b X)^6] \leqslant \mathbf{E}[(a + bX)^2]^3$. The result is trivial for $a = 0$, and therefore we may assume that $a = 1$. If we expand out the powers, and use the fact that $\mathbf{E}[X^k] = 0$ for odd k, we find that the inequality is equivalent to

$$1 + 15\rho^2 b^2 + 15\rho^4 b^4 + \rho^6 b^6 \leqslant 1 + 3b^2 + 3b^4 + b^6$$

If $\rho^2 \leqslant 1/5$, the inequality holds. □

We can use similar strategies to show that X is $(2, 8, 1/\sqrt{7})$ hypercontractive, and so on. In fact, we can generalize this theorem to a much general calculation.

Theorem 5.10 ((p,2) hypercontractivity)**.** *If X is uniform on $\{\pm 1\}$, and $1 \leqslant p < 2$, then the inequality $\|a + b\rho X\|_2 \leqslant \|a + bX\|_p$ holds for all $\rho \leqslant \sqrt{p-1}$. Thus X is $(p, 2, \sqrt{p-1})$ hypercontractive.*

Proof. By homogeneity, it suffices to prove the theorem assuming either $a = 0$, or $a = 1$. We may also assume that $\rho = \sqrt{p-1}$, and that $|b| \leqslant 1$. It then suffices to show the result for $|b| < 1$, for the $|b| = 1$ case follows by continuity. Thus we want to show

$$\|1 + \sqrt{p-1}bX\|_2^p \leqslant \|1 + bX\|_p^p$$

which is the same as $\mathbf{E}[(1 + \sqrt{p-1}bX)^2]^{p/2} \leqslant \mathbf{E}[(1 + bX)^p]$, and since $\mathbf{E}[X] = 0$, that

$$(1 + (p-1)b^2)^{p/2} \leqslant \mathbf{E}[(1 + bX)^p]$$

But now we find using the inequality $(1 + t)^x \leqslant 1 + xt$ for $t \geqslant 0$, $x \in [0, 1]$, hence

$$(1 + (p-1)b^2)^{p/2} \leqslant 1 + \frac{p(p-1)}{2}b^2$$

and we can apply the binomial theorem. □

Using the conjugate norms, we can obtain $(2, q)$ hypercontractivity results from the result above, using the following general result.

Theorem 5.11. *If T is a self-adjoint operator on $L^2(X)$, and if $\|Tf\|_q \leqslant C\|f\|_p$ holds for all f, then $\|Tg\|_{p'} \leqslant C\|g\|_{q'}$ holds for all g.*

Proof. We use the fact that $L_p(X)^* = L_q(X)$, and Hölder's inequality to conclude that

$$\|Tg\|_{p'} = \sup_{\|f\|_p = 1} \langle f, Tg\rangle = \sup_{\|f\|_p = 1} \langle Tf, g\rangle \leqslant \sup_{\|f\|_p = 1} \|Tf\|_q \|g\|_{q'} \leqslant C\|g\|_{q'}$$

and this completes the proof. □

Corollary 5.12. *A uniformly random bit is $(2, q, 1/\sqrt{q-1})$ hypercontractive.*

5.3 Two Function Hypercontractivity

Theorem 5.13. *If $f, g : \mathbf{B}^n \to \mathbf{B}$ are Boolean functions, and we pick constants $0 \leqslant r, s \leqslant 1$, $0 \leqslant \rho \leqslant \sqrt{rs} \leqslant 1$, then*

$$\mathbf{E}_{Y \sim N_\rho(X)}[f(X)g(Y)] \leqslant \|f\|_{1+r}\|g\|_{1+s}$$

Proof. We might as well assume $\rho = \sqrt{rs}$, and then we write

$$\mathbf{E}_{Y \sim N_\rho(X)}[f(X)g(Y)] = \mathbf{E}[T_{\sqrt{r}} f \, T_{\sqrt{s}} g] \leqslant \|T_{\sqrt{r}} f\|_2 \|T_{\sqrt{s}} g\|_2 \leqslant \|f\|_{1+r}\|g\|_{1+s}$$

where the last part of the inequality used $(p, 2)$ hypercontractivity. □

We have only proved $(p, 2)$ hypercontractivity for $n = 1$, so technically we have only proven this theorem for one dimensional functions, but surprisingly we can combine both of these results and the apply induction to obtain the result for all dimensions.

Theorem 5.14. *Let $0 \leqslant \rho \leqslant 1$, and suppose $\mathbf{E}_{Y \sim N_\rho(X)}[f(X)g(Y)] \leqslant \|f\|_p \|g\|_q$ holds for all choices of $f, g \in L^2(X)$. Then the inequality holds for all $f, g \in L^2(X^n)$.*

Proof. The proof is by induction on n. Given a function $f : \mathbf{B}^n \to \mathbf{R}$, and $x \in \mathbf{B}$, write f_x for the function $f_x : \mathbf{B}^{n-1} \to \mathbf{R}$ where $f_x(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, x)$. If X and Y are ρ correlation, then

$$\mathbf{E}[f(X)g(Y)] = \mathbf{E}_{X_n, Y_n}[\mathbf{E}[f_{X_n}(X) g_{Y_n}(Y)]] \leqslant \mathbf{E}_{X_n, Y_n}[\|f_{X_n}\|_p \|f_{Y_n}\|_q]$$

If we write $F(x) = \|f_x\|_p$, and $G(y) = \|f_y\|_q$, two Boolean functions, then we may apply the $n = 1$ case to conclude

$$\mathbf{E}[f(X)g(Y)] \leqslant \mathbf{E}_{X_n, Y_n}[F(X_n)G(Y_n)] \leqslant \|F\|_p \|G\|_q$$

And we calculate $\|F\|_p = \|f\|_p$, $\|G\|_q = \|g\|_q$, completing the proof. □

Chapter 6

Boolean Functions and Gaussians

In this chapter, we discuss how we can understand Boolean functions by approximating them by continuous probability distributions on the real numbers. The advantage of working over continuous distributions is that the geometry of the real numbers is at our disposal, so we can use differentiation and rotational symmetry to understand our problem in greater detail. Some techniques of Boolean functional analysis generalize to the space of continuous probability distributions on $\mathbf{R}^n$. However, we can no longer use a uniform distribution to construct structures on the space, because $\mathbf{R}^n$ is non-compact. The standard alternative is to use the Gaussian distribution $N(0,1)$, defined by the probability density function

$$f(x) = \frac{e^{-(x_1^2+\cdots+x_n^2)/2}}{(2\pi)^{n/2}}$$

We will let $L^p(\mathbf{R}^n)$ denote the space of functions with finite pth moment *with respect to the Gaussian distribution*, rather than the Lebesgue measure. In this chapter we will assume, unless otherwise specified, that an arbitrary random variable X over the real numbers is normally distributed (with mean zero and variance one). Thus if we have a function $g \in L^1(\mathbf{R}^n)$, then $\mathbf{E}[g(X)]$ will be assumed to be taken with respect to the normal distribution, and in particular $\|g\|_p = \mathbf{E}[|g(X)|^p]^{1/p}$.

Let us begin by generalizing the definition of stability to Gaussian integrable functions. If X and Y are normally distributed real-valued random variables, we say they are ρ correlated if $\mathbf{E}[XY] = \rho$. This uniquely defines the distribution of (X, Y), which is just the normal distribution with mean

zero and with covariance matrix

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

We can obtain a ρ correlated random variable Y by letting $Y = X$ with probability ρ, and letting Y be equal to a normal distribution independent of X with probability $1-\rho$. More rigorously, if we define random variables A and Z, such that X, Z, and A are all mutually independent, where Z is normally distributed and A is a Bernoulli random variable with $\mathbf{P}(A = 1) = \rho$, and if we let $Y = AX + (1-A)Z$, then Y is normally distributed, and

$$\mathbf{E}[XY] = \rho\mathbf{E}[X^2] + (1-\rho)[E](XZ) = \rho$$

Another way to construct two ρ correlated random variables from a pair $Z = (Z_1, Z_2)$ of independent normal distributions is to find a pair of unit vectors $u, v \in \mathbf{R}^2$ with $\langle u, v \rangle = \rho$, and then to define $X = \langle u, Z \rangle = u_1 Z_1 + u_2 Z_2$, $Y = v_1 Z_1 + v_2 Z_2$. Then $X, Y \sim N(0,1)$, and

$$\mathbf{E}[XY] = u_1 v_1 \mathbf{E}[Z_1^2] + u_2 v_2 \mathbf{E}[Z_2^2] = \langle u, v \rangle = \rho$$

so ρ correlated vectors are obtained from projecting an independent normal distribution onto two vectors at an angle $\arccos(\rho)$ from each other.

If X and Y are random vectors, we will say they are ρ correlated if the coordinates of the vectors are independent of each other, and X_i is ρ correlated to Y_i for each index i. Given $x \in \mathbf{R}^n$, we will write $Y \sim N_\rho(x)$ if $Y \sim N(\rho x, 1-\rho^2)$, which means exactly the Y is identically distributed to $\rho x + \sqrt{1-\rho^2}X$, where $X \sim N(0,1)$. If Y is ρ correlated to X, then $(Y|X = x) \sim N_\rho(x)$. Conversely. if Y is a variable with $(Y|X = x) \sim N_\rho(x)$, then Y is normally distributed and ρ correlated to X. We often write $Y \sim N_\rho(X)$ if Y is ρ correlated to X.

The discrete definition of stability for Boolean functions is defined by

$$\mathrm{Stab}_\rho(f) = \mathbf{E}_{Y \sim N_\rho(X)}[f(X)f(Y)]$$

But since we have now defined all the terms in the equation above for Gaussian-integrable functions, this means we can simply let this formula define the stability of functions on Gaussian space. In place of the Bonami-Beckner operator T_ρ, we have the **Ornstein-Uhlenbeck** operators U_ρ, also known as the **Gaussian noise operator**, defined by

$$(U_\rho f)(x) = \mathbf{E}_{Y \sim N_\rho(x)}[f(Y)] = \mathbf{E}\left[f\left(\rho x + \sqrt{1-\rho^2}X\right)\right]$$

and we have then justified that $\mathrm{Stab}_\rho(f) = \langle f, U_\rho f\rangle$.

Theorem 6.1. *U_ρ is a contraction operator on $L^p(\mathbf{R}^n)$, for any $1 \leqslant p \leqslant \infty$.*

Proof. For $p = \infty$, we find

$$\Big|(U_\rho f)(x)\Big| = |\mathbf{E}[f(X)]| \leqslant \|f\|_\infty$$

hence $\|U_\rho f\|_\infty \leqslant \|f\|_\infty$. Otherwise, we apply the convexity of $t \mapsto t^p$ and Jensen's inequality to conclude that

$$\begin{aligned}\|U_\rho f\|_p^p = \mathbf{E}_X[(U_\rho f)(X)^p] &= \mathbf{E}_X\left[\mathbf{E}_{Y\sim N_\rho(X)}[f(Y)]^p\right] \\ &\leqslant \mathbf{E}_{X,Y\sim N_\rho(X)}[f(Y)^p] \\ &= \|f\|_p^p\end{aligned}$$

because if $Y \sim N_\rho(X)$, in the sense that $(Y|X = x) \sim N_\rho(x)$ for each x, then Y is normally distributed, and ρ correlated with X. □

We think of U_ρ as a 'smoothing operator' on the set of Gaussian integrable functions, the same as for the noise operator T_ρ, but here we actual can measure how smooth the resulting function is.

Theorem 6.2. *If $f \in L^1(\mathbf{R}^n)$, and $-1 < \rho < 1$, then $U_\rho f$ is C^∞.*

Proof. Denoting $\sqrt{1-\rho^2}X$ as Y, we find

$$(U_\rho f)(x) = \mathbf{E}[f(\rho x + Y)] = (f * f_Y)(\rho x)$$

Since f_Y is C^∞, this implies $U_\rho f$ is as well. □

Theorem 6.3. *For any $f \in L^1(\mathbf{R}^n)$, as $\rho \to 1$, $U_\rho f \to f$.*

Proof. First assume f is uniformly continuous. Then for any ε, there is δ such that $|f(x+h) - f(x)| < \varepsilon$ for $|h| < \delta$. If ρ is chosen large enough that if $Y \sim N(\rho x, \sqrt{1-\rho^2})$, then $\mathbf{P}(|Y - x| < \delta) \geqslant 1 - \varepsilon$, then

$$\begin{aligned}\|U_\rho f - f\|_1 &= \mathbf{E}\Big[|(U_\rho f)(X) - f(X)|\Big] \\ &= \mathbf{E}\Big[|\mathbf{E}_{Y\sim N_\rho(X)}[f(Y)] - f(X)|\Big] \\ &= \mathbf{P}(|Y - X| < \delta)\mathbf{E}[|f(Y) - f(X)|\,||Y - X| < \delta] \\ &+ \mathbf{P}(|Y - X| \geqslant \delta)\mathbf{E}[|f(Y) - f(X)|\,||Y - X| \geqslant \delta] \\ &\leqslant \varepsilon + \varepsilon\mathbf{E}[|f(Y) - f(X)|\,||Y - X| \geqslant \delta]\end{aligned}$$

If we let $\delta \to 0$, then we can apply the monotone convergence theorem to conclude that

$$\begin{aligned}\mathbf{E}\left[|f(Y)-f(X)|\,|\,|Y-X| \geqslant \delta\right] &= \frac{1}{\mathbf{P}(|Y-X| \geqslant \delta)} \int_{|Y-X| \geqslant \delta} |f(Y)-f(X)| \\ &\to \mathbf{E}[|f(Y)-f(X)|]\end{aligned}$$

Hence for ρ large enough, we find $\|U_\rho f - f\|_1 \leqslant \varepsilon + \varepsilon \mathbf{E}[|f(Y)-f(X)|]$, and we may let $\varepsilon \to 0$ to conclude that $\|U_\rho f - f\|_1 \to 0$. □

We also obtain hypercontractivity results for the the Ornstein-Uhlenbeck operator, by reducing the problem to the case of Boolean functions.

Theorem 6.4. *Consider $f, g \in L^1(\mathbf{R}^n)$, and suppose $0 \leqslant \rho \leqslant \sqrt{rs} \leqslant 1$, then*

$$\langle f(X), (U_\rho g)(X)\rangle = \langle (U_\rho f)(X), g(X)\rangle \leqslant \|f\|_{1+r}\|g\|_{1+s}$$

Proof. Given a series $X_1, \ldots, X_m$ of uniformly random binary strings in $\mathbf{B}^n$, construct ρ correlated random binary strings $X_1', \ldots, X_m'$, and then consider the real-valued random variable

$$Y = \frac{X_1 + \cdots + X_m}{\sqrt{m}} \qquad Y' = \frac{X_1' + \cdots + X_m'}{\sqrt{m}}$$

The central limit theorem implies that Y and Y' converges in distribution to $N(0,1)$, and (Y, Y') converges in distribution to a pair of ρ correlated normally distributed functions. But then as $m \to \infty$ we find that

$$\mathbf{E}[f(Y)g(Y')] \to \langle f(X), (U_\rho g)(X)\rangle$$

$$\|f(Y)\|_{1+r} \to \|f(X)\|_{1+r}$$

and thus it suffices to prove that

$$\mathbf{E}[f(Y)g(Y')] \leqslant \|f(Y)\|_{1+r}\|g(Y)\|_{1+s}$$

But this follows from the hypercontractivity result for Boolean functions, because we can view $f(Y)$ and $g(Y)$ as functions on $\mathbf{B}^{nm}$. □

We used an inequality of this form to derive a hypercontractivity result for the noise operator on Boolean functions, and we find that this inequality gives us an immediate hypercontractivity result about the Gaussian noise operation, which is proved in the same way as the result is proved for Boolean functions.

Corollary 6.5. *If $1 \leqslant p \leqslant q \leqslant \infty$, and $0 \leqslant \rho \leqslant \sqrt{\frac{p-1}{q-1}}$, then $\|U_\rho f\|_q \leqslant \|f\|_p$.*

6.1 Hermite Polynomials

We desire an orthonormal basis for the space of square Gaussian integrable functions $L^2(\mathbf{R})$ which diagonalizes the noise operators U_ρ. First, we note that the space of polynomials is dense in $L^2(\mathbf{R})$, so that $1, x, x^2, \ldots$ is a Schauder basis for $L^2(\mathbf{R})$. We can therefore apply the Gram Schmidt orthogonalization process to this bases to obtain an orthogonal basis (without normalizing). The basis $H_1(x), H_2(x), \ldots$ obtained is known as the set of **Hermite polynomials**. To obtain these polynomials, we require knowledge of the quantities $\langle x^i, x^j \rangle = \mathbf{E}[x^{i+j}]$. These are the *moments* of the Gaussian distribution, and the standard technique for computing these values is to calculate the moment generating function of the distribution. If $X \sim N(0,1)$, then the Moment generating function is the transform $\varphi(t) = \mathbf{E}[e^{tX}]$, and provided this function is well defined in an open interval around the origin, it is differentiable, and $\varphi^{(n)}(0) = \mathbf{E}[X^n]$. In particular, if $X \sim N(0,1)$, then the moment generating function is $\varphi(t) = e^{t^2/2} = \sum_{k=0}^{\infty} (t^2/2)^k / k!$, so that

$$\mathbf{E}[X^{2n}] = (2n)!/(2^n n!) \qquad \mathbf{E}[X^{2n+1}] = 0$$

The coefficients of the power series expansion of this function then give us the moments of the Gaussian distribution. Thus in particular we can explicitly calculate that

$$H_0(x) = 1 \quad H_1(x) = x \quad H_2(x) = x^2 - 1 \quad H_3(x) = x^3 - 3x$$

However, for theoretical work there is a more powerful method to calculate all the polynomials at once. Note that if X and Y are ρ-correlated, then (X, Y) is identically distributed to $(\langle v, Z \rangle, \langle w, Z \rangle)$, where v and w are 2-dimensional unit vectors with $\langle v, w \rangle = \rho$, and Z is a 2-dimensional nor-

mal distribution. Thus we may calculate

$$\begin{aligned}
\mathbf{E}[e^{tX+sY}] &= \mathbf{E}[e^{(tv_1+sw_1)Z_1+(tv_2+sw_2)Z_2}] \\
&= \mathbf{E}[e^{(tv_1+sw_1)Z_1}]\mathbf{E}[e^{(tv_2+sw_2)Z_2}] \\
&= \exp\left(\frac{(tv_1+sw_1)^2}{2}\right)\exp\left(\frac{(tv_2+sw_2)^2}{2}\right) \\
&= \exp\left(\frac{t^2\|v\|^2+2ts\langle v,w\rangle+s^2\|w\|^2}{2}\right) \\
&= \exp\left(\frac{t^2+2\rho ts+s^2}{2}\right)
\end{aligned}$$

It thereby follows that

$$\sum_{n=0}^{\infty}\frac{\rho^n}{n!}t^n s^n = e^{\rho ts} = \frac{\mathbf{E}[e^{tX+sY}]}{\exp\left(\frac{t^2+s^2}{2}\right)} = \mathbf{E}\left[\exp\left(tX-(t^2/2)\right)\exp\left(sY-(s^2/2)\right)\right]$$

And if we consider the expansion

$$\exp\left(tX-(t^2/2)\right) = \sum_{n=0}^{\infty}\frac{P_n(X)}{n!}t^n$$

we find that

$$\sum_{n=0}^{\infty}\frac{\rho^n}{n!}t^n s^n = \sum_{n,m=0}^{\infty}\frac{\mathbf{E}[P_n(X)P_m(Y)]}{n!m!}t^n s^m$$

Hence

$$\mathbf{E}[P_n(X)P_m(Y)] = \begin{cases} n!\rho^n & n=m \\ 0 & n\neq m \end{cases}$$

In particular, if we let $Y = X$ (so Y is 1-correlated to X), we find the P_n are orthogonal polynomials in X, and $\|P_n(X)\|_2^2 = n!$, so $P_n(X)/\sqrt{n!}$ is an orthogonal set in $L^2(\mathbf{R})$. By observation, we see that the $P_n(X)$ are monic polynomials, so they are actually the Hermite polynomials!

Now to apply Fourier analysis, it is useful to normalize the Hermite polynomials so they form an orthonormal basis over $L^2(\mathbf{R})$. We will denote the normalized Hermite polynomials by h_n. From our calculations,

if $f \in L^2(\mathbf{R})$ has a Hermite expansion $f(x) = \sum a_n h_n(x)$, then $\mathbf{E}[f(X)] = a_0$, $\mathbf{V}[f(X)] = \sum_{n \neq 0} a_n^2$, and $U_\rho f = \sum \rho^n a_n h_n(x)$, and so

$$\mathrm{Stab}_\rho(f) = \sum \rho^n a_n^2$$

We may obtain an orthogonal basis for $L^2(\mathbf{R}^n)$ by taking $n$ary products of the Hermite polynomials. If $\alpha = (\alpha_1, \ldots, \alpha_n)$ is a multi-index, we will let $h_\alpha = \prod_{k=1}^n h_{\alpha_k}$, and then if $f(x) = \sum_\alpha a_\alpha h_\alpha(x)$, then we find $\mathbf{E}[f(X)] = a_0$, $\mathbf{V}[f(X)] = \sum_{\alpha \neq 0} a_\alpha^2$, and $(U_\rho f)(x) = \sum \rho^{|\alpha|} a_\alpha f(x)$, so $\mathrm{Stab}_\rho(f) = \sum \rho^{|\alpha|} a_\alpha^2$.

6.2 Borell's Isoperimetric Theorem

The main aim of introducing Gaussian probability theory to the study of Boolean functions is to prove the majority is stablest theorem, which says that if $f : \mathbf{B}^n \to [-1, 1]$ is unbiased, with $\mathrm{Inf}_i(f) \leqslant \tau$ for all indices i, then

$$\mathrm{Stab}_\rho(f) \leqslant 1 - \frac{2}{\pi} \arccos \rho + o_\tau(1)$$

If we are to believe this conjecture, then we must believe a corollary for Gaussian distribution. If $g : \mathbf{R} \to [-1, 1]$ is a real-valued function, then if we take uniformly distributed independant random variables $X_1, \ldots, X_n$ over $\mathbf{B}$, then we can view

$$g_n(X) = g\left(\frac{X_1 + \cdots + X_n}{\sqrt{n}}\right)$$

as a Boolean function on $\mathbf{B}^n$, and the central limit theorem implies that $\mathrm{Stab}_\rho(g_n) \to \mathrm{Stab}_\rho(g)$. If g is unbiased, then $\mathbf{E}[g_n(X)] \to 0$, and so

$$\mathrm{Stab}_\rho(g_n) \leqslant 1 - (2/\pi) \arccos \rho + o_\tau(1) + \mathbf{E}[g_n(X)]$$

and if we take $n \to \infty$, we should conclude that $\mathrm{Stab}_\rho(g) \leqslant 1 - (2/\pi) \arccos \rho$. This result is true, and is known as **Borell's Isoperimetric Theorem**.

Theorem 6.6. *Fix $\rho \in (0, 1)$. For any Gaussian square integrable $f : \mathbf{R}^n \to [-1, 1]$ with $\mathbf{E}[f(X)] = 0$, $Stab_\rho(f) \leqslant 1 - (2/\pi) \arccos \rho$.*

If we expect the Majority is stablest theorem to hold for Boolean functions, Borell's isoperimetric theorem must be provable. But we shall soon find that we can reduce the Boolean majority is stablest theorem to the isoperimetric theorem, so that the two theorems are essentially equivalent.

There is an equivalent way to state the isoperimetric theorem. First, the convexity of the functional $f \mapsto \|U_\rho f\|_2$ for $\rho \geqslant 0$ implies that we need only prove the result for functions $f : \mathbf{R}^n \to \mathbf{B}$ in order for the theorem to hold in general. In this case, we can interpret f as identifying some subset A of the plane, with $f(x) = 1$ if $x \in A$, and $f(x) = -1$ if $x \notin A$. We define the **Gaussian volume** of A, to be $\gamma(A) = \mathbf{P}(X \in A)$, where X is normally distributed (γ is really the measure by which we define the Gaussian distribution). The isoperimetric theorem then says that for any $\theta \in (0, \pi/2)$, and a subset A of $\mathbf{R}^n$ with $\gamma(A) = (1/2)$, then for any $\cos\theta$ correlated normal random variables X and Y, the **Rotation Sensitivity** $\mathbf{RS}_A(\cos\theta)$ of A has a bound

$$\mathbf{RS}_A(\cos\theta) = \mathbf{P}(\mathbf{I}(X \in A) \neq \mathbf{I}(Y \in A)) \geqslant \theta/\pi$$

though the theorem is true for all θ, we will show how to prove the theorem whenever $\theta = \pi/2n$ for some positive integer n. The key result to proving this is that

Lemma 6.7. *For any subset A, the function $\mathbf{RS}_A$ is subadditive.*

Proof. s □

Corollary 6.8. *The Borell isoperimetry theorem holds for $\theta = \pi/2n$.*

Proof. It is easy to see that $\mathbf{RS}_A(\pi/2) = 1/2$ for any set A, but this implies that

$$1/2 = \mathbf{RS}_A(\pi/2) = \mathbf{RS}_A(n(\pi/2n)) \geqslant n\mathbf{RS}_A(\pi/2n)$$

and this gives the inequality exactly. □

6.3 The Invariance Theorem

Now we've reduced problems about Gaussian functions to Boolean functions, we want to reverse this strategy, and prove facts about Boolean functions through Gaussian functions. The correspondence mapping Gaussian

functions to Boolean functions is easy, because we can obtain real valued functions from bits $X_1, \ldots, X_n \in \mathbf{B}^n$ by considering their accumulation $(X_1 + \cdots + X_n)/\sqrt{n} \in \mathbf{R}$. But the converse is less trivial, because it seems strange to try to apply real values to a function $f : \mathbf{B}^n \to \mathbf{R}$. However, the key result of these entire notes is that any such function f can be written as $f(x) = \sum a_S x^S$, and since x^S is just a monomial in the variables $x_1, \ldots, x_n$, it is easy just to 'plug real values in'. A natural question is how similar the two random quantities we obtain are. The general result we will obtain in this line of thinking is the 'invariance theorem', giving bounds on similarity for arbitrary degree polynomials. The theorem for degree one polynomials is used throughout probability theory.

Theorem 6.9 (Berry-Esseen)**.** *Let $X_1, \ldots, X_n$ be independent random variables with $\mathbf{E}[X_i] = 0$, $\mathbf{V}(X_i) = \sigma_i^2$ and assume $\sum \sigma_i^2 = 1$. If we let $S = \sum X_i$, and let $Z \sim N(0,1)$, then for all $t \in \mathbf{R}$,*

$$|\mathbf{P}(S \leqslant t) - \mathbf{P}(Z \leqslant t)| \leqslant c \sum \|X_i\|_3^3$$

where c is a universal constant slightly less than 0.56*.*

We will prove the Berry-Esseen theorem using a method which extends to derive the general invariance principle, known as the 'replacement method'. Instead of trying to show that $\sum X_i \approx Z$, we will show that $\sum X_i \approx \sum Z_i$, where the Z_i are independent normally distributed random variables with mean zero and variance σ_i^2. This is essentially the same statement, because when we sum normal distributions with mean zero we obtain a normal distribution which varies according to the sum of the variances of the component distributions. And now we see that the replacement method doesn't really need to use Gaussian distributions at all, because the Berry-Essen theorem says that if $X_1, \ldots, X_n$ are independent random varables, as are $Y_1, \ldots, Y_n$, with $\mathbf{E}[X_i] = \mathbf{E}[Y_i]$ and $\mathbf{E}[X_i^2] = \mathbf{E}[Y_i^2]$, then $S_X = \sum X_i \approx \sum Y_i = S_Y$. The degree to how similar these two random variables are depends on them being 'reasonable' in some sense, which corresponds to the error bound in the theorem above.

Now classically, the Berry Esseen theorem bounds the difference between the cumulative distribution functions of S_X and S_Y, but we can also consider other tests of similarity. For instance, is $\|S_X\|_1 \approx \|S_Y\|_1$, or is the average Hausdorff distance to some set A the same, i.e. is $\mathbf{E}[d(S_X, A)] \approx \mathbf{E}[d(S_Y, A)]$. We can generalize all these conditions to showing that $\mathbf{E}[\psi(S_X)] \approx$

$\mathbf{E}[\psi(S_Y)]$ for some 'test function' ψ. Our generalization of the Berry Esseen theorem will be able to prove upper bounds on how similar these estimates are, but will require ψ to be a function in C^3. If the function isn't in C^3, it can always be approximated by a C^3 function, and the ability to approximate the function will result in a different constant in the approximation, just like in the standard Berry Essen theorem.

Theorem 6.10 (The Invariance Theorem)**.** *Let $X_1,;X_n, Y_1,\ldots,Y_n$ be independent random variables with matching 1st and 2nd moments. Then for any C^3 function ψ,*

$$|\mathbf{E}[\psi(S_X)] - \mathbf{E}[\psi(S_Y)]| \leqslant \frac{\|\psi'''\|_\infty}{6} \sum \|X_i\|_3^3 + \|Y_i\|_3^3$$

Proof. For each $0 \leqslant m \leqslant n$, define the 'replacement' random variable

$$H_m = Y_1 + \cdots + Y_m + X_{m+1} + \cdots + X_n$$

By the triangle inequality, since $H_0 = S_X$, and $H_n = S_Y$, we find that

$$|\mathbf{E}[\psi(S_X) - \psi(S_Y)]| \leqslant \sum_{m=1}^{n} |\mathbf{E}[\psi(H_{m-1}) - \psi(H_m)]|$$

and so it suffices to show that

$$\begin{aligned}\frac{\|\psi'''\|_\infty}{6}(\mathbf{E}|X_m|^3 + \mathbf{E}|Y_m|^3) &\geqslant |\mathbf{E}[\psi(H_{m-1}) - \psi(H_m)]] \\ &= |\mathbf{E}[\psi(U_m + X_m) - \psi(U_m + Y_m)]\end{aligned}$$

where $U_m = Y_1 + \cdots + Y_{m-1} + X_{m+1} + \cdots + X_n$ is a random variable independent of Y_m and X_m.

Now we can probably expect X_m and Y_m to be rather small compared to U_m, so it is probably wise to apply Taylor's theorem, which tells us that

$$\psi(u + \delta) = \psi(u) + \delta\psi'(u) + \frac{\psi''(u)}{2}\delta^2 + \frac{\psi'''(u^*)}{6}\delta^3$$

for some $u^* \in [u, u + \delta]$. Applying this to $\psi(U_m + X_m)$ and $\psi(U_m + Y_m)$, we find random variables U_m^*, U_m^{**} such that

$$\psi(U_m + X_m) = \psi(U_m) + \psi'(U_m)X_m + \frac{\psi''(U_m)}{2}X_m^2 + \frac{\psi'''(U_t^*)}{6}X_m^3$$

$$\psi(U_m + Y_m) = \psi(U_m) + \psi'(U_m)Y_m + \frac{\psi''(U_m)}{2}Y_m^2 + \frac{\psi'''(U_t^{**})}{6}Y_m^3$$

Applying this to the inequality we were trying to prove, we reduce the inequality to analyzing the expression

$$\left|\mathbf{E}[\psi'(U_m)(X_m - Y_m)] + \mathbf{E}\left[\frac{\psi''(U_m)}{2}(X_m^2 - Y_m^2)\right] + \mathbf{E}\left[\frac{\psi'''(U_m^*)X_m^3 - \psi'''(U_m^{**})Y_m^3}{6}\right]\right|$$

But we find that since U_m is independent of X_m and Y_m, that

$$\mathbf{E}[\psi'(U_m)(X_m - Y_m)] = \mathbf{E}[\psi'(U_m)]\mathbf{E}[X_m - Y_m] = 0$$
$$\mathbf{E}[\psi''(U_m)/2(X_m^2 - Y_m^2)] = \mathbf{E}[\psi''(U_m)/2]\mathbf{E}[X_m^2 - Y_m^2] = 0$$

and so we are really only bounding

$$\frac{1}{6}\left|\mathbf{E}\left[\psi'''(U_m^*)X_m^3 - \psi'''(U_m^{**})Y_m^3\right]\right| \leqslant \frac{\|\psi'''\|_\infty}{6}\left(\mathbf{E}[|X_m|^3] + \mathbf{E}[|Y_m|^3]\right)$$

and this completes the proof. □

Most of the time, our functions ψ will not be differentiable, but the space of differentiable functions is always dense in the set of all functions, so we always try approximating this function by differentiable functions to obtain proper approximations.

Lemma 6.11. *If ψ is c-Lipschitz, and $\eta > 0$ is arbitrary, then there is a function $\tilde{\psi}$ with $\|\psi - \tilde{\psi}\|_\infty \leqslant \eta c$ and $\|\tilde{\psi}^{(k)}\| = O_k(c/\eta^{k-1})$.*

Theorem 6.12. *Let ψ be c-Lipschitz. Then*

$$|\mathbf{E}[\psi(S_X) - \psi(S_Y)]| \leqslant O(c)\left(\sum \|X_i\|_3^3 + \|Y_i\|_3^3\right)^{1/3}$$

Proof. Taking $\tilde{\psi}$ as in the last lemma, we find that

$$|\mathbf{E}[\tilde{\psi}(S_X) - \tilde{\psi}(S_y)]| \leqslant O(c/\eta^2)\left(\sum \|X_i\|_3^3 + \|Y_i\|_3^3\right)$$

But $\|\tilde{\psi} - \psi\|_\infty \leqslant c\eta$ implies that

$$|\mathbf{E}[\tilde{\psi}(S_X) - \psi(S_X)]| \leqslant \|\tilde{\psi} - \psi\|_\infty \leqslant c\eta$$

and similarily for S_Y. Thus

$$|\mathbf{E}[\psi(S_X) - \psi(S_Y)]| \leqslant O(c)\left(\eta + (1/\eta^2)\left(\sum \|X_i\|_3^3 + \|Y_i\|_3^3\right)\right)$$

choosing η to be minimize the result gives the required bound. □

We end this discussion by noting that the Invariance theorem is sometimes stronger than the Berry Essen theorem, and sometimes weaker. If we let the $Y_1,\dots,Y_n$ be Gaussian in the lemma, we can obtain the standard Berry Essen theorem, but the resulting error term is on the order of $O((\sum\|X_i\|^3)^{1/4})$, rather than $O(\sum\|X_i\|^3)$, however, we do obtain a sharper constant in our new version of the theorem.

In the context of our 'substitution technique' for proving things about Boolean functions by reducing them to functions over Gaussian distributions, let us consider what we've just proved. Given independant variables $X_1,\dots,X_n$ and $Y_1,\dots,Y_n$ with $\mathbf{E}[X_i]=\mathbf{E}[Y_i]=0$, $\mathbf{V}[X_i]=\mathbf{V}[Y_i]=1$, we have shown that for any $a_1,\dots,a_n\in\mathbf{R}$, the distribution of $a_0+a_1X_1+\cdots+a_nX_n$ is approximately the same as $a_0+a_1Y_1+\cdots+a_nY_n$. Thus we have essentially shown that our 'substitution technique' works for degree one polynomials. In order to obtain complete results, we need to be able to show that $\sum a_S X^S$ is distributed the same as $\sum a_S Y^S$, and this will constitute the full Invariance theorem.

Theorem 6.13. *Suppose that $f(x)=\sum a_S X^S$ is a polynomial of degree k, and $X_1,\dots,X_n,Y_1,\dots,Y_n$ are independant 9-reasonable random variables with $\mathbf{E}[X_i]=\mathbf{E}[Y_i]=0$, $\mathbf{E}[X_i^2]=\mathbf{E}[Y_i^2]=1$, $\mathbf{E}[X_i^3]=\mathbf{E}[Y_i^3]=0$. If ψ is a C^4 real valued function, then*

$$|\mathbf{E}[\psi(f(X))-\psi(f(Y))]|\leqslant\frac{9^k}{12}\|\psi''''\|_\infty\sum Inf_i(f)^2$$

Proof. The proof is very similar to the other invariance theorem, except we'll use a 3rd order taylor expansion. For each n, define

$$U_m=(E_mf)(Y_1,\dots,Y_{m-1},\cdot,X_{m+1},\dots,X_m)$$
$$\Delta_m=(D_mf)(Y_1,\dots,Y_{m-1},\cdot,X_{m+1},\dots,X_n)$$

so that if we let $H_m=f(Y_1,\dots,Y_m,X_{m+1},\dots,X_n)$, then $H_{m-1}=U_m+\Delta_mX_m$ and $H_m=U_m+\Delta_mY_m$. We telescope the difference above like in the last invariance theorem to reduce the theorem of showing that

$$|\mathbf{E}[\psi(H_{m-1})-\psi(H_m)]|\leqslant\frac{9^k}{12}\|\psi''''\|_\infty\mathrm{Inf}_i(f)^2$$

If we take a 3rd order taylor expansion of $\psi(U_m+\Delta_mX_m)$ and $\psi(U_m+\Delta_mY_m)$, and then take the expectation over their difference, we find that

the 0th, 1st, 2nd, and 3rd moments cancel out, and we can upper bound the 4th moment difference to conclude that

$$\mathbf{E}[\psi(H_{m-1}) - \psi(H_m)] \leqslant \frac{\|\psi''''\|_\infty}{24}(\mathbf{E}[(\Delta_m X_m)^4] + \mathbf{E}[(\Delta_m Y_m)^4])$$

and we can upper bound the fourth moments using the Bonami lemma. Since $\Delta_m X_m$ has degree at most k, we find that the random variable is reasonable, and so

$$\mathbf{E}[(\Delta_m X_m)^4] \leqslant 9^k \mathbf{E}[(\Delta_m X_m)^2]^2 = 9^k \mathrm{Inf}_m(f)^2$$

which follows because $\Delta_m X_m = (L_m f)(Y_1, \ldots, Y_{m-1}, X_m, \ldots, X_n)$, where $(L_m f)(x) = \sum_{m \in S} a_S x^S$, and so if we let $Z = (Y_1, \ldots, Y_{m-1}, X_m, \ldots, X_n)$, then by using the independence of the X and Y, and the fact that the second moment of all variables is 1, we conclude that

$$\mathbf{E}[(L_m f)(Y_1, \ldots, Y_{m-1}, X_m, \ldots, X_n)^2] = \sum_{m \in S,T} a_S a_T \mathbf{E}[Z^{S \Delta T}] = \sum_{m \in S} a_S^2 = \mathrm{Inf}_m(f)$$

this completes the proof of the inequality. □

In particular, this theorem shows that understanding probabilistic facts about a Boolean polynomial $f(x) = \sum a_S x^S$ can be approached either by looking at $f(X)$, where X is a vector of uniformly random $\mathbf{B}$ bits, or where X is a vector of variance one Gaussian distributions. This, in tandem with the isoperimetric theorem for Gaussian functions, will be all that is required to prove the Majority is stablest theorem.

As with the one-dimensional invariance theorem, we can replace differentiable test functions ψ with Lipschitz functions, and we obtain a slightly worse bound.

Theorem 6.14. *If ψ is c Lipschitz, $\mathbf{V}(f) \leqslant 1$, and $Inf_i(f) \leqslant \varepsilon$ for all i, then $|\mathbf{E}[\psi(f(X)) - \psi(f(X))]| \leqslant O(c) 2^k \varepsilon^{1/4}$.*

The proof is essentially the same as in the invariance theorem. We require a little bit more discussion for majority is stablest.

Theorem 6.15. *Let $f : \mathbf{B}^n \to \mathbf{R}$ have $\mathbf{V}(f) \leqslant 1$. Let $k \geqslant 0$ and suppose $f^{\leqslant k}$ have ε small influences. Then for any c Lipschitz ψ we have*

$$\left|\mathbf{E}_{X \sim \mathbf{B}^n} \psi(f(X)) - \mathbf{E}_{X \sim N(0,1)} \psi(f(X))\right| \leqslant O(c)(2^k \varepsilon^{1/4} + \|f^{>k}\|_2)$$

In particular, if $h : \mathbf{B}^n \to \mathbf{R}$ *have* $\mathbf{V}(h) \leqslant 1$ *and no* (ε, δ) *notable coordinates then*

$$\left|\mathbf{E}_{X \sim \mathbf{B}^n} \psi(f(X)) - \mathbf{E}_{X \sim N(0,1)} \psi(f(X))\right| \leqslant O(c)\varepsilon^{\delta/3}$$

Proof. Write $f = f^{\leqslant k} + f^{>k}$, then use the triangle inequality and the Lipschitz property of ψ to find

$$\begin{aligned}&|\mathbf{E}[\psi(f^{\leqslant k}(X) + f^{>k}(x)) - \mathbf{E}[\psi(f^{\leqslant k}(Y) - f^{>k}(Y))]] \\ &\leqslant |\mathbf{E}\psi(f^{\leqslant k}(X)) - \mathbf{E}\psi(f^{\leqslant k}(Y))| + c\mathbf{E}|f^{>x}(X)| + c\mathbf{E}|f^{>k}(Y)|\end{aligned}$$

The first quantity is at most $O(c)2^k\varepsilon^{1/4}$. Cauchy Schwarz bounds the two values by $\|f^{>k}\|_2$ giving us the first inequality. Now given the second statement, let $g = T_{1-\delta}(f)$. Then $\mathbf{V}(g) \leqslant 1$ and $g^{\leqslant k}$ has ε small influences for any k, and $\|g^{>k}\|_2^2 \leqslant (1-\delta)^{2k}\mathbf{V}(f) \leqslant (1-\delta)^{2k} \leqslant e^{-2k\delta}$. Now we apply the first theorem, and optimize over k to obtain the required theorem. □

6.4 Majority is Stablest

We are now ready to prove the Majority is stablest theorem, by using the invariance principle to pass into Gaussian space twice.

Theorem 6.16. *Let* $f : \mathbf{B}^n \to [0,1]$. *Suppose that* $\mathbf{MaxInf}(f) \leqslant \varepsilon$, *or more generally, that* f *has no* $(\varepsilon, 1/\log(1/\varepsilon))$ *notable coordinates. Then for any* $0 \leqslant \rho < 1$,

$$\mathbf{Stab}_\rho(f) \leqslant \Lambda_\rho(\mathbf{E}[f]) + O(\log\log(1/\varepsilon)/\log(1/\varepsilon))\frac{1}{1-\rho}$$

Proof. Let $g = T_{1-\delta}f$, where we will find that letting $\delta = 3\log\log(1/\varepsilon)/\log(1/\varepsilon)$ is convinent. Assume that ε is sufficiently small, so that $0 < \delta, 1/20$. Note that $\mathbf{E}[g] = \mathbf{E}[f]$, and that

$$\mathbf{Stab}_\rho[g] = \sum \rho^{|S|}(1-\delta)^{2|S|}\widehat{f}(S)^2 = \mathbf{Stab}_{\rho(1-\delta)^2}(f)$$

But using the Lipschitz bound we found, we find that

$$|\mathbf{Stab}_{\rho(1-\delta)^2}(f) - \mathbf{Stab}_\rho(f)| \leqslant (\rho - \rho(1-\delta)^2)\frac{\mathbf{V}(f)}{1-\rho} \leqslant \frac{2\delta}{1-\rho}$$

This can be absorbed into the error term with our choice of δ, so it suffices to prove the theorem for g. Let $F : \mathbf{R} \to \mathbf{R}$ be the function with $F(x) = x^2$

for $x \in [0,1]$, and $F(x) = 1$ for $x \notin [0,1]$. Then F is 2-Lipschitz. This implies that

$$|\mathbf{E}_{X\sim\mathbf{B}^n}[F(T_{\sqrt{\rho}}g(X)) - \mathbf{E}_{X\sim N(0,1)^n}[F(T_{\sqrt{\rho}}h(X))]] \leqslant O(\varepsilon^{\delta/3}) = O\left(\frac{1}{\log(1/\varepsilon)}\right)$$

Note $T_{\sqrt{\rho}}g = T_{(1-\delta)\sqrt{\rho}}f$ is always in $[0,1]$, so

$$F(T_{\sqrt{\rho}}g(x)) = \mathbf{E}(T_{\sqrt{\rho}}g(x))^2$$

and so $\mathbf{E}[F(T_{\sqrt{\rho}}g(X))] = \mathbf{Stab}_\rho(g)$. Furthermore $T_{\sqrt{\rho}}g(X)$ is the same as $U_{\sqrt{\rho}}g(X)$. Thus it suffices to show that

$$|\mathbf{E}[F(U_{\sqrt{\rho}}g(X)) - \Lambda_\rho(\mathbf{E}[g])]| \leqslant O(1/\log(1/\varepsilon))$$

Define $G : \mathbf{R}^n \to [0,1]$ by letting $G(x) = 0$ if $g(x) < 0$, $G(x) = g(x) \in [0,1]$, and $G(x) = 1$ if $g(x) > 1$, and $G(x) = x$ otherwise. We shall show that

$$\left|\mathbf{E}[F(U_{\sqrt{\rho}}g(X)) - F(U_{\sqrt{\rho}}(F\circ G\circ g)(X))]\right| \leqslant O(1/\log(1/\varepsilon))$$

$$|\mathbf{E}[F(U_{\sqrt{\rho}}G(X))]| \leqslant \Lambda_\rho(\mathbf{E}[g]) + O(1/\log(1/\varepsilon))$$

These inequalities both follow from the fact that

$$\mathbf{E}[|g(X) - (G\circ g)(X)] \leqslant O(1/\log(1/\varepsilon))$$

The first inequalitiy follows from this inequality because F is 2-Lipschitz, so

$$\begin{aligned}\left|\mathbf{E}[F(U_{\sqrt{\rho}}g(X)) - F(U_{\sqrt{\rho}}(F\circ G\circ g)(X))]\right| &\leqslant 2\mathbf{E}[|U_{\sqrt{\rho}}g(X) - U_{\sqrt{\rho}}(G\circ g)(X)] \\ &\leqslant 2\mathbf{E}|g(X) - (G\circ g)(X)| \\ &\leqslant O(1/\log(1/\varepsilon))\end{aligned}$$

The second inequality follows because $U_{\sqrt{\rho}}(G\circ g)$ is valued in $[0,1]$ since $(G\circ g)$ is, hence

$$\mathbf{E}[F(U_{\sqrt{\rho}}(G\circ G)(X))] = \mathbf{E}[(U_{\sqrt{\rho}}(G\circ g)(X))^2] = \mathrm{Stab}_\rho(G\circ g) \leqslant \Lambda_\rho(\mathbf{E}[(G\circ g)(X)])$$

where this proof used the Isoperimetric theorem. But now we can estabilsh the second inequality because $|\mathbf{E}[(G\circ g)(X) - g(X)]| \leqslant O(1/\log(1/\varepsilon))$, and Λ_p is 2-Lipschitz.

Thus it remains to show that

$$\mathbf{E}[|g(X) - (G \circ g)(X)] \leqslant O(1/\log(1/\varepsilon))$$

and this follows becase the difference between the two terms above is $\mathrm{dist}_{[0,1]}$, and we can use the corollary to the invariance theorem to find

$$\left|\mathbf{E}_{X \sim N(0,1)}[\mathrm{dist}_{[0,1]}(g(X))] - \mathbf{E}_{X \sim \mathbf{B}^n}\mathrm{dist}_{[0,1]}(g(X))\right| \leqslant O(\varepsilon^{\delta/3}) = O(1/\log(1/\varepsilon))$$

But $\mathbf{E}[\mathrm{dist}_{[0,1]}(g(X))] = 0$ because $g(X) = T_{1-\delta}g$ is in $[0,1]$ always. $\square$

Chapter 7

Hypercontractivity Lecture

Bibliography

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

[2] Ryan O'Donnell *Analysis of Boolean Functions*