Università degli Studi di Pisa
Dipartimento di Informatica

Dottorato di Ricerca in Informatica
Università di Pisa - Udine

Ph.D. Thesis: TD-2/98

Mathematical Techniques for the Analysis of Boolean Functions

Anna Bernasconi

Abstract Any attempt to find connections between mathematical properties of functions and their computational complexity has a strong relevance to theory of computation. Indeed, there is the hope that developing new mathematical techniques could lead to discovering properties that might be responsible for lower bounds. The current situation is that none of the known techniques has yet led to lower bounds in general models of computation. The subject of this thesis is related to the above general arguments, especially to developing mathematical techniques for analyzing small depth circuits, via the investigation of combinatorial and computational properties of Boolean functions. To this aim, we have considered different mathematical areas, i.e., *abstract harmonic analysis*, *algebraic graph theory* and *algebraic geometry*, and have analyzed the extent to which certain mathematical techniques from these areas can provide some insight in our current understanding of Boolean function complexity. The main results of this thesis consist of new lower bounds on the complexity of explicit functions, exactly derived by applying the above techniques.

March 1998

ADDR: Corso Italia 40, 56126 Pisa, Italy. TEL: +39-50-887111. FAX: +39-50-887226
E-MAIL: bernasconi@imc.pi.cnr.it

Preface

Any attempt to find connections between mathematical properties of functions and their computational complexity has a strong relevance to theory of computation. Indeed, there is the hope that developing new mathematical techniques could lead to discovering properties that might be responsible for lower bounds. The current situation is that none of the known techniques has yet led to lower bounds in general models of computation.

The subject of this thesis is related to the above general arguments, especially to developing mathematical techniques for analyzing small depth circuits, via the investigation of combinatorial and computational properties of Boolean functions. To this aim, we have considered different mathematical areas, i.e., *abstract harmonic analysis*, *algebraic graph theory* and *algebraic geometry*, and have analyzed the extent to which certain mathematical techniques from these areas can provide some insight in our current understanding of Boolean function complexity. The main results of this thesis consist of new lower bounds on the complexity of explicit functions, exactly derived by applying the above techniques.

The first part of the thesis is devoted to a systematic study of the connections between Boolean function complexity and *abstract harmonic analysis* on the hypercube. We give an interpretation of harmonic analysis on the hypercube in terms of its *Laplacian Matrix*, which allows us to describe in a natural way the interplay between Fourier coefficients and the *sensitivity* of Boolean functions, a measure of how the value of the function is sensitive to changes in the input. We also show the relationship between the Fourier coefficients and the *influences* of the variables.

We then use the Fourier based mathematical techniques to find connections between structural and combinatorial properties of Boolean functions and complexity.

We study the connections between *average sensitivity* and sparseness, and use them to find properties of symmetric AC^0 functions. We analyze the long standing open problem regarding the relationship between maximal sensitivity and *block sensitivity*, an extension of the sensitivity which gives a full characterization of the time needed to compute a Boolean function on a PRAM (Parallel Random Access Machine) with an unlimited number of processors. Because of the implications regarding spectral (i.e., mathematical) properties on the one side, and parallel complexity on the other side, it is very important to analyze the relationship between these two complexity measures. As a starting point for attacking this intricate combinatorial problem, we consider *average* versions of these measures, and show that, in the average case, there is no polynomial (or even exponential) relationship between them. We investigate the representations of Boolean functions as polynomials over R and over the field $GF(2)$, and prove that the degree of the polynomial representation of a Boolean function over the field $GF(2)$ is equal to the size of its largest 'odd subfunction'. We then present some consequences of this result regarding combinatorial and spectral properties of *read-once formulas* and of *odd* functions.

We then introduce the notions of *balanced* and *strongly balanced Boolean functions* and examine the complexity of these functions using the Fourier based techniques. These results are applied to derive new lower bounds on the size complexity of previously unclassified Boolean functions. These bounds generalize the well known result stating that constant depth circuits require exponential size to compute the parity function.

We also use tools from *algebraic graph theory* to analyze Boolean functions. We indeed show that the Fourier analysis of Boolean functions can be viewed as a Cayley graph eigenvalue problem. We import results from algebraic graph theory to analyze Boolean functions with a few distinct Fourier coefficients, and show that the fewer the number of distinct coefficients is, the stronger are the algebraic properties of the function. Using this tool, we also find a nice interpretation for the well known class of *bent functions* in terms of *strongly regular graphs*.

The last part of the thesis is devoted to the analysis of the *algebraic geometry approach* to Boolean function complexity. The idea of this approach is to bring the investigation on Boolean function complexity into the framework of algebraic geometry, and to use mathematical tools as Gröbner bases and Hilbert functions for complexity purposes. In [65] Smolensky showed that the values taken by the Hilbert function of the ideals associated to monotone Boolean functions can be used to derive lower bounds on the degree of the polynomial representation of these functions. In this thesis we continue the investigation started by Smolensky, and provide a complete characterization of the Hilbert function of the ideals associated to symmetric Boolean functions. We then apply this characterization to analyze the behavior of the majority function on symmetric subsets of the Boolean cube, and to derive some nontrivial structural properties of the Hilbert function itself.

Acknowledgments

I would like to thank Milvio Capovani and Bruno Codenotti, who followed my work providing technical support and constant encouragement.

My thanks go also to Lavinia Egidi, with whom it has always been a pleasure working.

I spent a year at The University of Chicago. It was for me a great experience to work there and enjoy the warm hospitality and the excellent environment of the Computer Science Department. Special thanks go to Janos Simon for inviting me to visit and for his useful advice.

Finally, I want to thank all the people at the I.M.C. and the I.E.I. (third floor, old wing) institutes of the Italian National Research Council.

Contents

Chapter 1

Introduction

The goal of Complexity Theory is to characterize the amount of resources needed for the computation of functions by establishing lower bounds on their inherent computational difficulty. The effort to achieve this goal has been only partly successful. Except in a few special circumstances, we have been unable to demonstrate that particular problems are computationally difficult, even though there are many which appear to be so. The most fundamental questions, such as the famous **P** versus **NP** question which simply asks whether it is harder to find a proof that to check one, remain far beyond our present abilities.

One of the reasons of the difficulty in proving that problems have high complexity seems to lie in the lack of mathematical techniques to prove lower bounds for general models of computation. Therefore, any attempt to find connections between mathematical properties and complexity has a strong relevance to this field.

This thesis is intended to be a step in this direction. We are interested in developing mathematical techniques for proving lower bounds on the Boolean circuit model of computation. We consider different mathematical areas, abstract harmonic analysis, algebraic graph theory and algebraic geometry, and analyze the extent to which mathematical techniques from these areas can give additional insight in our current understanding of Boolean functions.

Our main objectives are to:

- find conditions that link complexity properties of Boolean functions to mathematical properties by using mathematical tools such as the Fourier transform;
- find connections between combinatorial properties of Boolean functions and complexity;
- reach a better understanding of the relations between the different measures of complexity of Boolean functions.

We would expect several potential benefits from such a study:

- definition of interesting new complexity classes defined by properties of the Fourier transforms of the functions in the class;
- new lower bound proofs;
- a better understanding of combinatorial properties of Boolean functions.

The results of the thesis indicate that our approach can be fruitful.

1.1 The Boolean Circuit Model of Computation

As of today, no nonlinear lower bounds have been proved on general models of computation for any language in **NP**. This apparent difficulty of proving nonlinear lower bounds has led people to approach the problem in an indirect way, with the following line of attack:

- study a weaker model of computation, better suited to lower bounds proofs;
- develop mathematical techniques for proving lower bounds in this weaker model; and
- try to extend these techniques to the general situation.

This is precisely the framework that has led many researchers to study the field of *Circuit Complexity*, which regards the attempts to prove lower bounds on the *Boolean Circuit* model of computation.

Let us define a general Boolean circuit and point out some well known facts concerning circuits as a computational model.

A *Boolean circuit* is a directed, acyclic graph, whose nodes are divided into *input, operation* and *output* nodes. Input nodes have in-degree 0 and are labeled with a variable x_i or a constant 0 or 1. Operation nodes (or *gates*) have in-degree $k > 0$ and are labeled by Boolean operations from a given basis. The output node has in-degree 1 and out-degree 0, and is labeled by a different variable symbol.

We refer to the in-degree of a node as is *fan-in* and its out-degree as its *fan-out*. We will only consider circuits over the *canonical basis*, consisting of the AND, OR and NOT operations. It is well known that the canonical basis is a complete one: any Boolean function can be expressed by means of these operations.

Without loss of generality we can assume that negations in the circuit occur only as negated input variables. If negations appear higher up in the circuit we can move them down to the inputs using De Morgan's laws. This at most doubles the number of gates of the circuit. Finally, observe that for unbounded fan-in circuits we have alternating levels of AND and OR gates, since two adjacent gates of the same type can be collapsed into one gate.

A Boolean circuit computes a function in a natural way: a circuit with n input nodes *computes* a function $f : \{0,1\}^n \rightarrow \{0,1\}$ if for any assignment of values from

$\{0,1\}$ to the input variables $x_1, \ldots, x_n$, the value read from the output node equals the value of the function $f(x_1, \ldots, x_n)$ on that assignment.

Since a circuit has a fixed structure, it can only solve problems of fixed size. Therefore, a single circuit cannot be considered a reasonable model of computation, but we must take into consideration *families of circuits*, one for each length of the input strings.

A family of circuits $\{C_n\}_{n \in N}$ computes a function $g : \{0,1\}^* \rightarrow \{0,1\}$ if and only if, for any $n \in N$, C_n computes g_n, where g_n is g restricted to the input strings of length n.

Some researchers impose a *uniformity condition* of the circuits of the family, which allows them to finitely describe the generic element of the family, once the length n of the input is specified. Precisely, we say that a family $\{C_n\}_{n \in N}$ is *uniform* if and only if there exists an algorithm that, on input n, generates a description of C_n. Specializations of the notion of uniformity can be obtained by appropriately bounding the amount of resources available for circuit generation.

We will work with nonuniform families of circuits, which means that we make no assumptions about the difficulty of constructing a circuit of a family, once the length n of the input is specified. Hence the lower bounds we prove will be as strong as possible.

The main complexity measures for a circuit are its *size* and its *depth*. The size of a circuit is defined as the total number of operation nodes; the depth is the length of the longest directed path from an input to the output node.

Let $f : \{0,1\}^n \rightarrow \{0,1\}$. Then, the *size complexity* (or *circuit complexity*) of f is the size of the smallest circuit computing f. The concepts of size and depth can be extended to families of circuits. Let $g : \{0,1\}^* \rightarrow \{0,1\}$, and $h : N \rightarrow N$. We say that g has size complexity h if for all n the size of the smallest circuit computing g_n is equal to $h(n)$, where g_n is g restricted to the input strings of length n. A language is a subset of $\{0,1\}^*$, and the size complexity of a language is that of its characteristic function.

It is not too hard to prove that any problem that can be solved in time $T(n)$ on a Turing machine can be solved by circuits of size $O(T(n) \log T(n))$. Consequently lower bounds on the size of general circuits computing a function f yield corresponding lower bounds on the time complexity of f on a Turing machine.

Circuits are an attractive model for proving lower bounds for several reasons. As we have already mentioned, they are closely related in computational power to Turing machines. Moreover, among computational models the circuit model has an especially simple definition and so may be more amenable to combinatorial analysis. Indeed, even though nonuniform circuits are even more powerful than Turing machines, they seem better suited to lower bounds proofs since they do not contain hard to control features, like moving heads and changing states. But in spite of this general feeling, no nonlinear lower bounds have been proved yet on the size of general circuits computing an explicit function.

By a simple counting argument, it is not too hard to prove that almost every Boolean function of n variables requires circuits of size $\Omega(2^n/n)$; but for explicit functions the best bounds known are only linear.

By weakening the power of the circuit, however, very interesting results can be obtained. One way to limit the power of the model is to study *monotone circuits*. Monotone circuits contain AND and OR gates, but no negations, which restricts the functions that can be computed to be monotone: i.e. changing an input from 0 to 1 cannot change the value of the function from 1 to 0. Razborov [57] proved superpolynomial lower bounds on the size of monotone circuits computing the NP-complete clique function. The results were improved by Alon and Boppana [2] to give exponential lower bounds for the clique function and by Andreev [4], who obtained exponential lower bounds for other languages in NP.

Another way to limit the power of the Boolean circuits model of computation is that of studying *small depth circuits*. These circuits have the complete instruction set of negations, AND and OR gates with arbitrarily many inputs, but the depth is restricted to be small, either a constant independent of the input length, or a slowly growing function of the input length.

The problem of proving lower bounds for small depth circuits has attracted the attention of many researchers in the field. Functions typically considered have been simple functions such as *parity* and *majority*. Parity is defined to be the sum modulo two of the input bits, and majority is defined to be 1 when at least half of the input bits are 1. Furst, Saxe and Sipser [33] and Ajtai [1] were the first to demonstrate a superpolynomial lower bound for constant-depth circuits computing the parity function. Subsequently Yao [75], by giving a deeper analysis of the method of Furst, Saxe and Sipser, was able to give a much sharper exponential lower bound. Håstad [35] further strengthened and simplified this argument, obtaining near optimal bounds.

Some complexity classes have been defined on the model of small depth circuits.

The class NC^k is defined as the class of Boolean function families that can be computed by uniform sequences of constant fan-in circuits in polynomial size and depth $O((\log n)^k)$. Moreover, we have

$$NC = \bigcup_{k=0}^{\infty} NC^k .$$

NC can be described as the class of problems that are efficiently computable in parallel (indeed, the depth of a Boolean circuit can be viewed as a measure of the parallel time needed to compute a function).

Similarly, the class AC^k is defined as the class of Boolean function families that can be computed by uniform sequences of unbounded fan-in circuits in polynomial size and depth $O((\log n)^k)$; and we have

$$AC = \bigcup_{k=0}^{\infty} AC^k .$$

Sometimes, these classes are defined without the uniformity condition.

For $k = 0$ we get the class AC^0, which is the class of functions that can be computed by polynomial size and constant depth circuits. More precisely, $AC^0[d]$ is the class of functions computable by an AC^0 circuit of depth d.

It is well known that $NC^k \subseteq AC^k \subseteq NC^{k+1}$ for every k, and that $NC = AC$. The only strict inclusions that are known are $NC^0 \neq AC^0 \neq NC^1$. The first one is trivial, since AC^0 functions may depend on all inputs while this is impossible for NC^0 functions. The second inclusion follows from the lower bounds for parity circuits since parity is in NC^1. Note that this result actually represents one of the few meaningful separation results between complexity classes proved up to now.

Other interesting complexity classes over small depth circuits are the classes $ACC[m]$, which are natural extensions of the notion of AC^0. The idea is to increase the number of basic operations, and try to obtain lower bounds for more powerful circuit models. We consider the following operation: for any integer $m \in N$, let MOD_m be the Boolean function which outputs 1 if the number of ones in the input string is congruent to 0 (mod m), and outputs 1 otherwise. $ACC[m]$ is then defined to be the class of functions with AND, OR and negation gates, plus any finite set of MOD_m operation nodes. We also have

$$ACC = \bigcup_{m \in N} ACC[m]\,.$$

Several results about $ACC[m]$ circuits have been obtained by applying mathematical techniques based on the polynomial representation of Boolean functions. For instance, it has been proven that the MOD_q function cannot be computed by an $ACC[p]$ circuit, where p and q are two distinct prime numbers (see [58] and [64]). For $q = 2$ we get $parity \equiv MOD_2 \notin ACC[p]$; and, since $parity \in NC^1$, we then have $ACC[p] \subset NC^1$, for any prime $p \neq 2$. The complexity classes $ACC[p]$ are the largest classes known to be strictly contained in NC^1.

1.2 Content and Results of the Thesis

In this thesis we are interested in developing mathematical techniques for proving lower bounds on small depth circuits, and therefore in studying combinatorial and computational properties of Boolean functions.

We consider three different mathematical areas: *abstract harmonic analysis*, *algebraic graph theory* and *algebraic geometry*.

Abstract harmonic analysis. With respect to the first area, we study the connections between Boolean function complexity and *Abstract Harmonic Analysis* on the hypercube.

The *Fourier transform* of a Boolean function is an invertible linear mapping of the values of the function onto a set of coefficients, known as Fourier coefficients. This transformation is such that the Fourier coefficients contain information about the *regularities* of the function, and thus about its computational complexity. Thus, our goal is to look at the *frequency domain* of Boolean functions in order to get complexity properties.

The main part of the thesis is actually devoted to a systematical study of the properties of the Fourier analysis on the hypercube, with the aim of gaining new insights for the analysis of Boolean functions. Indeed, we present a complete framework in which to analyze the Fourier Transform of Boolean Functions.

In addition to a review of known results, we give an interpretation of the harmonic analysis on the hypercube in terms of its *Laplacian Matrix*, which allows us to describe in a natural way the interplay between Fourier coefficients and the *sensitivity* of Boolean functions, a measure of how the value of the function is sensitive to changes in the input. We also show the relationship between the Fourier coefficients and the *influences* of the variables.

We study the connections between *average sensitivity*, sparseness and complexity, and use them to find properties of symmetric AC^0 functions. We characterize the Fourier transform of the function composition $f(g_1, \cdots, g_k)$ in terms of the Fourier coefficients of f and of the g_i, by means of a matrix V, whose entries can be explicitly given in terms of the convolution of the Fourier spectra of the g_i. We hope that this will eventually enable us to prove lower bound theorems using a 'top-down' approach.

We define some new complexity classes, essentially composed of functions that have most of their power spectrum in a relatively narrow contiguous band. Such classes are analogous to the class AC^0 (it is well known [45] that AC^0 has such a property). We show that, like AC^0, they are learnable in quasi-polynomial time, under the uniform distribution. We also derive a version of Khrapchenko's theorem expressed in terms of Fourier coefficients.

We then use the Fourier based mathematical techniques to find connections between structural and combinatorial properties of Boolean functions and complexity. The main results of the thesis actually come from this analysis.

We analyze the long standing open problem regarding the relationship between maximal sensitivity and *block sensitivity*, an extension of the sensitivity which gives a full characterization of the time needed to compute a Boolean function on a PRAM (Parallel Random Access Machine) with an unlimited number of processors. Because of the implications regarding spectral (i.e., mathematical) properties on the one side, and parallel complexity on the other side, it is very important to analyze the relationship between these two complexity measures. Rubinstein [60] exhibits an infinite family of Boolean functions with a quadratic gap between sensitivity and block sensitivity. This is the best known lower bound. It is a long-standing open problem whether the gap may be bigger, e.g. superpolynomial. As a starting point for attacking this intricate combinatorial problem, we consider *average* versions of these measures, and show that for the family of functions considered by Rubinstein, the average block sensitivity is $\Omega(\sqrt{n})$, while the average sensitivity is asymptotically zero. Thus, in the average case, there is no polynomial (or even exponential) relationship between sensitivity and block sensitivity.

We investigate the representations of Boolean functions as polynomials over R and over the field $GF(2)$, and prove an interesting characterization theorem about the degree over $GF(2)$. Precisely, we prove that the degree of the polynomial representation of a Boolean function over the field $GF(2)$ is equal to the size of its largest odd subfunction. We then present some consequences of this result regarding combinatorial and spectral properties of *read-once formulas*, i.e. formulas in which each variable appears only once, and of *odd* functions, i.e. functions that take the value 1 on an odd number of input strings.

We finally present our main result related to this topic: we show how the Fourier based techniques can be applied to derive new lower bounds on the size complexity of previously unclassified Boolean functions. These bounds generalize the well known result stating that constant depth circuits require exponential size to compute the parity function.

We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *balanced* if it takes the value 1 on exactly half of the input strings; *balanced up to the level* k (shortly k*-balanced*) if any subfunction depending on k variables is balanced; and *strongly balanced* if f is balanced up to a level k such that $n-k = \omega\left((\log n)^d\right)$ for a constant d. Our main result is that AC^0-circuits cannot compute strongly balanced functions. More precisely, we prove that a circuit of constant depth d require size $\Omega\left(2^{\frac{(n-k)^{1/d}}{20}}\right)$ to compute a k-balanced function. Note that this class of functions contains the parity and its complement as very special cases, since they are the only functions balanced up to $k = 1$. The main tool of the proof is the harmonic analysis on the hypercube, that yields an interesting spectral characterization of k-balanced functions, together with the result proved in [45], stating that AC^0 functions have almost all of their *power spectrum* on the low-order coefficients.

Algebraic graph theory. The interest in the connections between *algebraic graph theory* and Boolean function complexity comes from the following observation: the Fourier analysis of Boolean functions can be viewed as a Cayley graph eigenvalue problem. Indeed, this observation allows us to use tools from algebraic graph theory for investigations related to the Fourier coefficients of Boolean functions. In particular, we import results from algebraic graph theory to analyze Boolean functions with a few distinct Fourier coefficients. We show that the fewer the number of distinct coefficients is, the stronger are the algebraic properties of the function, and find a nice interpretation for the well known class of *bent functions* in terms of *strongly regular graphs*.

Algebraic geometry. As we have already pointed out, some results about small depth circuits have been obtained by applying mathematical techniques based on the polynomial representation of Boolean functions. One of the goals of this approach is that of showing that Boolean functions computable by certain kinds of circuits can be represented by polynomials of a given type, e.g. low degree polynomials. These results have naturally provided lower bound techniques.

The polynomial representation makes it possible to bring the investigation on Boolean function complexity into the framework of *algebraic geometry.* In fact, a Boolean function f on n variables can be represented as a real multivariate polynomial $p(x_1, , \dots , x_n)$ whose zero set corresponds to the set of zeroes of f. Then the affine variety defined by

- $p(x_1, , \dots , x_n) = 0,$
- $x_i^2 - x_i = 0\,,\;\; i = 1, \dots , n,$

describes the zeroes of f, together with the proper restrictions on the variables (i.e. $x_i \in \{0, 1\},\ 1 \leq i \leq n$). This variety can be represented in terms of *Gröbner bases* and its dimension investigated in terms of the *Hilbert function*, whose value can be used for complexity purposes. More precisely, in [65] Smolensky showed that the values taken by the Hilbert function are related to certain lower bounds, that follow from the comparison of the dimension of the above variety with the dimension of the variety defined by a low degree polynomial.

In this thesis, we continue the investigation started by Smolensky, and provide a complete characterization of the Hilbert function of the ideals associated to symmetric Boolean functions. We then apply this characterization to analyze the behavior of the majority function on symmetric subsets of the Boolean cube, and to derive some nontrivial structural properties of the Hilbert function itself.

One of the objectives of such a study is to give additional insight about the limit of the information content of the Hilbert function and thus about the extent to which the investigation of Boolean function complexity can take advantage of algebraic geometry tools.

Outline of the thesis. The chapters of this thesis are organized as follows. In Chapter 2 we give the necessary background on abstract harmonic analysis and study the Fourier representation of Boolean functions. We then present some applications of Fourier Analysis to Circuit Complexity. Chapter 3 presents the long standing open problem regarding the relationship between maximal sensitivity and block sensitivity. Chapter 4 is devoted to the presentation of our main results: we prove a characterization theorem about the degree of a Boolean function over $GF(2)$; we present some properties of odd functions and prove a theorem regarding the degree of Boolean functions computable by read-once formulas; we then derive a lower bound on the size complexity of k-balanced Boolean functions. Chapter 5 describes the connections between algebraic graph theory and Boolean function complexity. Chapter 6 explores the application of certain algebraic geometry techniques involving Hilbert functions and Gröbner bases to the analysis of properties of Boolean functions and gives some results and applications for symmetric functions. Finally, Chapter 7 offers some concluding remarks and indicates some directions for future research.

The main results of this thesis appeared in the open literature in [10], [11], [12], [13], [14], [15] and [16]. Portions of the thesis are based on joint works with Bruno Codenotti, Lavinia Egidi and Janos Simon.

1.3 Related Work

In this section, we briefly review the main literature about the topics discussed in this thesis.

Fourier techniques in the Boolean setting were well known in the late 60's. They were used as a tool for analysis and synthesis of switching circuits, especially as a tool for sensitivity analysis and error detection. A review article of 1971 [44] illustrates the applications of Fourier analysis to the theory of switching circuits. Chapter 2 in [61] is dedicated to the description of some mathematical background on error detection in digital machines. In particular, the *Boolean difference* function - which is the difference between the value of a function at a given argument w and the value at an argument which is a Hamming neighbor of w - is used to analyze error propagation. Although without real justification, Karpovsky [39] has proposed to use the number of nonvanishing Fourier coefficients of a Boolean function f as a measure of its complexity.

While most of these results were forgotten, Fourier techniques reappeared in the 80's. Hurst et al. [36] relate the circuit complexity of a Boolean function to its power spectrum coefficients. Brandman et al. [20] establish a relationship between the Fourier coefficients of a Boolean function f and (i) the average size of any decision tree for f; (ii) the minimum number of $\wedge$ gates in a circuit computing f according to its disjunctive normal form. Kahn et al. [38] find connections between the influence of variables and harmonic analysis and use theorems on influence to prove results on rapidly mixing Markov chains. In addition, they relate the average sensitivity of functions to their Fourier coefficients. However this relation was already implicit in the work of [36]. Ben-Or and Linial [9] study collective coin flipping, where the collective coin is viewed as a Boolean function. In this case, measuring the influence of a subset of variables onto the value of the function corresponds to measuring how the collective coin is sensitive to the presence of faults and the goal is to find Boolean functions on which the influence of each variable is as small as possible, to prevent a small subset of variables, e.g. the set of faulty processors, taking control of the collective coin. Linial et al. [45] take advantage of the relation between the average sensitivity of Boolean functions and their Fourier transform to prove several facts, e.g., that sets in AC^0 have low average sensitivity (see Section 2.7 for more details). Bruck [21] and Bruck and Smolensky [22] use abstract harmonic analysis to derive necessary and sufficient conditions for a Boolean function to be a polynomial threshold. Finally, in recent years, several results in Learning Theory have been obtained by applying Fourier-based techniques (see [50] and [37] for a comprehensive and exhaustive survey on this topic).

With respect to the analysis of the connections between algebraic graph theory and Boolean function complexity, the work developed in this thesis represents, to our knowledge, the first approach to using tools from spectral theory of graphs for investigations related to the Fourier coefficients.

Finally, the algebraic geometry approach to Boolean function complexity has been developed by Smolensky, who has shown that the values taken by the Hilbert function of the ideals associated to Boolean functions, are responsible for certain lower bounds (see [65]). These lower bounds are mainly related to complexity measures such as the degree of the polynomial representation of Boolean functions. A comprehensive and exhaustive survey on the polynomial representation of Boolean functions can be found in [8].

1.4 Notation and Basic Definitions

In this section we provide some of the notation we use throughout the thesis, and recall some basic definitions.

Unless otherwise specified, the indexing of vectors and matrices starts from 0 rather than 1. A^T denotes the transpose of a matrix A. $\rho(B)$ denotes the spectral radius of a matrix B, i.e. the largest of the absolute values of the eigenvalues of B.

Given a Boolean function f on n binary variables, we will use different kinds of notation: the *classical notation*, where the input string is given by n binary variables; the *set notation*, based on the correspondence between the set $\{0,1\}^n$ and and the power set of $\{1,2,\ldots,n\}$; the 2^n tuple *vector representation* $f = (\, f_0\, f_1 \,\ldots f_{2^n-1}\,)$, where $f_i = f(b(i))$ and $b(i)$ is the binary expansion of i.

We will use the notation $|f|$ to denote the number of strings accepted by the function f, i.e. $|f| = |\{w \in \{0,1\}^n \mid f(w) = 1\}|$.

Given a binary string $w \in \{0,1\}^n$, we denote with $w^{(i)}$ the string obtained from w by flipping its i-th bit ($1 \le i \le n$), i.e. w and $w^{(i)}$ differ only on the i-th bit. In the same way, given a subset of indices $B \subseteq \{1,2,\ldots,n\}$, we denote with $w^{(B)}$ the string w with all bits in B flipped, i.e. $w^{(B)}$ differs from w exactly on those bits whose index belongs to B. Furthermore $|w|$ denotes the number of ones in the binary string w, which is sometimes called *cardinality* of the string because of the correspondence between sets of positive integers and strings over the alphabet $\{0,1\}$.

If w and v are two binary strings of the same length, then $d(w,v)$ and $w \oplus v$ denote their Hamming distance and the string obtained by computing the *exclusive or* of the bits of w and v. Moreover, $w^T v$ denotes the inner product of the two strings.

Finally, all the logarithms are to the base 2, and the notation *polylogn* stands for a function growing like a polynomial in the logarithm of n.

We now review some basic definitions.

Symmetric Boolean functions

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *symmetric* if its output only depends on the number of input bits which are equal to 1. Thus, for any permutation π of the set $\{1,2,\ldots,n\}$ we have

$$f(x_1, x_2, \ldots, x_n) = f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}).$$

Monotone Boolean functions

Let $v = (v_1, v_2, \ldots, v_n), w = (w_1, w_2, \ldots, w_n) \in \{0,1\}^n$. We use the canonical ordering, i.e. $v \leq w$ if and only if $v_i \leq w_i$ for all i, where $0 \leq 1$. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called *monotone* if $v \leq w$ implies $f(v) \leq f(w)$, for all $w, v \in \{0,1\}^n$.

Note that a Boolean function is monotone if and only if it can be computed by a monotone circuits, i.e. by a circuit which contains only AND and OR gates, but no negations.

Restriction

A *restriction* ρ is a mapping of the input variables to the set $\{0,1,\star\}$, where

- $\rho(x_i) = 0$ means that we substitute the value 0 for x_i;
- $\rho(x_i) = 1$ means that we substitute the value 1 for x_i;
- $\rho(x_i) = \star$ means that x_i remains a variable.

Given a function f on n binary variables, we will denote by f_ρ the function obtained from f by applying the restriction ρ; f_ρ will be a function of the variables x_i for which $\rho(x_i) = \star$.

The *domain* of a restriction ρ, $dom(\rho)$, is the set of variables mapped to 0 or 1 by ρ. The *size* of a restriction ρ, $size(\rho)$, is defined as the number of variables which were given the value $\star$, i.e. $size(\rho) = n - |dom(\rho)|$.

Random Restriction

A *random restriction* ρ with parameter p satisfies, independently for different x_i

- $\rho(x_i) = 0$ with probability $\frac{1}{2} - \frac{p}{2}$;
- $\rho(x_i) = 1$ with probability $\frac{1}{2} - \frac{p}{2}$;
- $\rho(x_i) = \star$ with probability p.

Observe that we have probability p of keeping a variable. Thus the expected number of variables remaining is pn.

Chapter 2

Fourier Analysis of Boolean functions

The Fourier transform of a Boolean function is an invertible linear mapping of the values of the function onto a set of coefficients, known as Fourier coefficients. This transformation is such that the Fourier coefficients contain information about the *regularities* of the function, and thus about its computational complexity. In this chapter we give the necessary background on abstract harmonic analysis and study the Fourier representation of Boolean functions. We then present some applications of Fourier Analysis to Circuit Complexity (more interesting results will be presented in Chapter 4).

This chapter is organized as follows. In Section 2.1 we give some background on abstract harmonic analysis and study the Fourier representation of Boolean functions. In Section 2.2 we introduce the notion of *average* and *maximal sensitivity* of Boolean functions, and show that the average sensitivity turns out to be a *submodular formal measure of complexity* (see [59] and [73]). We also give precise estimates for the distribution of average sensitivity for the set of all Boolean functions of a given number of variables. In Section 2.3 we give an interesting interpretation of the harmonic analysis on the hypercube in terms of the *Laplacian Matrix*. This leads to a natural view of the connections between the notions of average and maximal sensitivity on one side, and of Fourier coefficients on the other one. In Section 2.4 we relate the Fourier coefficients to the *influences* of sets of variables, and in Section 2.5 we show how to compute the Fourier coefficients and estimate the sensitivity of symmetric and monotone functions. In Section 2.6 we introduce the notion of *generalized sensitivity* (or k-sensitivity) and study its properties.

The second part of the chapter is devoted to the presentation of some applications of Fourier Analysis to Circuit Complexity. In Section 2.7 we review a result by Linial et al. (see [45]) about a spectral characterization of AC^0 functions. We also spend a few words on a very powerful tool to prove lower bounds on small depth circuits, the *Håstad's Switching Lemma*. In Section 2.8, we present some new results: using certain connections between sensitivity and sparsity of languages, we characterize spectrum and sensitivity of symmetric AC^0 functions (Section 2.8.1); we give some formulas expressing the Fourier coefficients of composition of Boolean functions (Section 2.8.2); we define some new complexity classes, essentially composed of functions that have most of their power spectrum in a relatively narrow contiguous band (Section 2.8.3); we finally derive a version of Khrapchenko's theorem expressed in terms of Fourier coefficients (Section 2.8.4).

Some of results presented in this chapter also appeared in [12], [13] and [15].

2.1 Fourier Transform of Boolean functions

We give some background on abstract harmonic analysis. Our main sources are [44] and [47].

The impatient reader may skip much of this section, and simply remember the idea of Fourier transforms. We consider Boolean functions as 0-1 valued real functions defined on the domain $\{0,1\}^n$. They are a vector space of dimension 2^n, and the set of functions

$$\{f_x(y) = \delta(x,y) \mid x \in \{0,1\}^n \},$$

is a basis. Another basis is given by the set of functions

$$\{g_B(y) = \sum_{i \in B} y_i \mid B \subseteq \{1,2,\ldots,n\} \},$$

where the sum is modulo 2. The Fourier coefficients of f are the coefficients of f in this second basis. Note that, a real (integer) valued function on $\{0,1\}^n$ will have a real (integer) valued transform.

We now give a more formal definition of the transform. The set $\mathcal{F}$ of all real functions on the hypercube $\{0,1\}^n$

$$\mathcal{F} = \{f : \{0,1\}^n \to \mathcal{R}\}$$

is a 2^n-dimensional real vector space with an inner product defined by

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{w \in \{0,1\}^n} f(w)g(w),$$

where $f, g \in \mathcal{F}$. The domain of $\mathcal{F}$ is a locally compact Abelian group and the elements of its range, i.e. 0 and 1, can be added and multiplied as complex numbers.

The above properties allow one to analyze $\mathcal{F}$ by using tools from harmonic analysis. This means that it is possible to construct an *orthogonal* basis set of *Fourier transform kernel functions* for $\mathcal{F}$. The kernel functions of the Fourier transform are defined in terms of a group homomorphism from $\{0,1\}^n$ to the direct product of n copies of the multiplicative subgroup $\{\pm 1\}$ on the unit circle of the complex plane.

Definition 2.1 *For any $w \in \{0,1\}^n$, the* Fourier transform kernel function *relative to w is defined as*

$$Q_w(x) = (-1)^{w_1 x_1}(-1)^{w_2 x_2} \dots (-1)^{w_n x_n} = (-1)^{w^T x},$$

where $x \in \{0,1\}^n$.

The functions Q_w are also known as *group characters.*

Theorem 2.1 ([44]) *The set of the 2^n kernel functions $\{Q_w | w \in \{0,1\}^n\}$ is an orthonormal basis for $\mathcal{F}$.*

Proof The Q_w are all distinct; for suppose $Q_u = Q_v$, although u and v differ in, say, the i-th coordinate. Then $u_i \neq v_i$ and $Q_u(e^{(i)}) \neq Q_v(e^{(i)})$, where $e^{(i)}$ is the string whose i-th bit is equal to 1, while all the other bits are equal to 0.

Since the 2^n functions Q_w are distinct, and $\mathcal{F}$ has dimension 2^n, they will span $\mathcal{F}$ if they are orthogonal. To prove orthogonality, we expand the inner product

$$\langle Q_w, Q_y \rangle = \frac{1}{2^n} \sum_x (-1)^{w^T x}(-1)^{y^T x} = \sum_x \prod_j (-1)^{(w_j + y_j) x_j} = \sum_x (-1)^{(w+y)^T x}.$$

Now $(w_j + y_j)$ has the range 0, 1, and 2 in each exponent; however, because $(-1)^{2x_j} = (-1)^0 = 1$, regardless of the value of x, the sum $(w_j + y_j)$ can be replaced by a modulo two sum for each j. Defining $u = w \oplus y$, we get

$$\langle Q_w, Q_y \rangle = \frac{1}{2^n} \sum_x (-1)^{u^T x},$$

where the sum is over all $x \in \{0,1\}^n$. To evaluate this inner product we consider two different cases. Let $\overline{0}$ denote the all zeroes string.

$\boldsymbol{w \neq y}$

In this case, we have $u = w \oplus y \neq \overline{0}$. Then, the orthogonality follows from the fact that the two sets $A = \{x \in \{0,1\}^n \mid u^T x = 1 \pmod 2\}$ and $B = \{x \in \{0,1\}^n \mid u^T x = 0 \pmod 2\}$ have the same cardinality, i.e. $|A| = |B| = 2^{n-1}$:

$$\begin{aligned} \langle Q_w, Q_y \rangle &= \frac{1}{2^n} \sum_x (-1)^{u^T x} = \frac{1}{2^n} \left(\sum_{x \in A} (-1)^1 + \sum_{x \in B} (-1)^0 \right) \\ &= \frac{1}{2^n} (-2^{n-1} + 2^{n-1}) = 0. \end{aligned}$$

$\boldsymbol{w = y}$

If $w = y$, we have $u = \overline{0}$. Thus, we get

$$\langle Q_w, Q_y \rangle = \frac{1}{2^n} \sum_x (-1)^{u^T x} = \frac{1}{2^n} \sum_x (-1)^0 = 1,$$

which proves orthonormality.

□

Using the set notation for Boolean functions, it is possible to give an interpretation to the meaning of the Fourier transform kernel functions. Let W be the subset associated to the string w, i.e. $W = \{i \in \{1, 2, \ldots, n\} \mid w_i = 1\}$. The kernel function Q_w can be written as follows

$$Q_w(x) = Q_W(x) = \prod_{i \in W} (-1)^{x_i} = (-1)^{\sum_{i \in W} x_i} .$$

Thus, $Q_W(x) = 1$ if the sum $\sum_{i \in W} x_i$ is even, while $Q_W(x) = -1$ if this sum is odd, and this means that the function Q_w is a measure of the parity of the string x over the set W.

We can now define the *Abstract Fourier Transform* of a Boolean function.

Definition 2.2 *Let $f : \{0,1\}^n \to \{0,1\}$. The* Abstract Fourier Transform *of f is the rational valued function $f^* : \{0,1\}^n \to Q$ which defines the coordinates of f with respect to the basis $\{Q_w(x), w \in \{0,1\}^n\}$, i.e.*

$$f^*(w) = \langle Q_w, f \rangle = \frac{1}{2^n} \sum_x Q_w(x) f(x) .$$

Then

$$f(x) = \sum_w Q_w(x) f^*(w)$$

is the Fourier expansion *of f.*

Using the binary 2^n-tuple representation for the functions f and f^*, and considering the natural ordering of the n-tuples x and w, one can derive a convenient matrix formulation for the transform pair. Let us consider a $2^n \times 2^n$ matrix H_n whose (i, j)-th entry h_{ij} satisfies

$$h_{ij} = (-1)^{b(i)^T b(j)} ,$$

where $b(i)^T b(j)$ denotes the inner product of the binary expansions of i and j, and $0 \le i, j \le 2^n - 1$. Thus, the i-th row of H_n coincides with the Fourier kernel function $Q_{\{i\}}$.

Note that the matrix H_n is the *Hadamard symmetric transform matrix* [44] and has the following property

$$H_n^{-1} = 2^{-n} H_n .$$

Moreover, it can be recursively defined as

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} , \qquad H_n = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix} .$$

If $f = [f_0\, f_1 \ldots f_{2^n-1}]^T$ and $f^* = \left[f_0^*\, f_1^* \ldots f_{2^n-1}^*\right]^T$, we get

$$\begin{aligned} f &= H_n\, f^* \\ f^* &= \frac{1}{2^n}\, H_n\, f\, . \end{aligned}$$

We now prove two identities, *autocorrelation identity* and *Parseval's identity*, which are a consequence of the orthogonality of the functions Q_w.

Lemma 2.1

Autocorrelation identity

$$\sum_v f(v)f(v \oplus u) = 2^n \sum_w Q_w(u)(f^*(w))^2\, .$$

Parseval's identity

$$\sum_v f(v) = 2^n \sum_w (f^*(w))^2\, .$$

Proof Let us first consider the Fourier expansion of f relative to the strings v and $v \oplus u$:

$$\begin{aligned} f(v) &= \sum_w (-1)^{w^T v} f^*(w)\, , \\ f(v \oplus u) &= \sum_\sigma (-1)^{\sigma^T (v \oplus u)} f^*(\sigma)\, . \end{aligned}$$

We have

$$\begin{aligned} \sum_v f(v)f(v \oplus u) &= \sum_v \sum_w \sum_\sigma (-1)^{w^T v} (-1)^{\sigma^T (v \oplus u)} f^*(w) f^*(\sigma) \\ &= \sum_v \sum_w \sum_\sigma (-1)^{w^T v} (-1)^{\sigma^T v} (-1)^{\sigma^T u} f^*(w) f^*(\sigma) \\ &= \sum_w \sum_\sigma (-1)^{\sigma^T u} f^*(w) f^*(\sigma) \sum_v (-1)^{w^T v} (-1)^{\sigma^T v}\, . \end{aligned}$$

From the orthogonality of the Fourier kernel basis

$$\sum_v (-1)^{w^T v} (-1)^{\sigma^T v} = 2^n \delta(w, \sigma)\, ,$$

we then obtain

$$\begin{aligned} \sum_v f(v)f(v \oplus u) &= 2^n \sum_w \sum_\sigma (-1)^{\sigma^T u} f^*(w) f^*(\sigma) \delta(w, \sigma) \\ &= 2^n \sum_w (-1)^{w^T u} (f^*(w))^2 = 2^n \sum_w Q_w(u)(f^*(w))^2\, . \end{aligned}$$

Parseval's identity can be derived from the autocorrelation identity, by putting $u = \overline{0}$ and by noting that over $\{0, 1\}$, $f^2(w) = f(w)$.

$\square$

2.1.1 Interpretation of the Fourier Coefficients

We now give an interpretation of the Fourier coefficients. We first need some definitions.

Definition 2.3 *The* order *of a Fourier coefficient $f^*(S)$ is the cardinality of the subset S.*

Definition 2.4 *The* degree *of a Boolean function f is the maximum order of the nonzero Fourier coefficients, i.e.*

$$deg(f) = \max\{\ |S| \mid f^*(S) \neq 0\ \}\,.$$

Note that this definition of degree coincides with the degree related to the polynomial representation of Boolean functions over the reals (see Section 4.1).

In the following, we show that the Fourier coefficients measure the correlation, under the uniform distribution, between the function and the parities of certain subsets of variables.

Zero-order coefficient

The zero-order Fourier coefficient corresponds to the empty set and it is equal to the probability that f takes the value 1, where the probability is taken with respect to the uniform distribution. Indeed, we have

$$f^*(\emptyset) = \frac{1}{2^n}\sum_w f(w)\,.$$

From Parseval's identity it then follows that

$$p = \Pr[f(w) = 1] = \frac{1}{2^n}\sum_w f(w) = f^*(\emptyset) = \sum_w (f^*(w))^2\,.$$

Coefficients of order k

The Fourier coefficients of order k, $0 < k \leq n$, $f^*(W)$, $|W| = k$, measure the correlation between the function and the parity of those variables w_j such that $j \in W$:

Proposition 2.1

$$f^*(W) = \frac{1}{2} - \Pr[f(w) = \bigoplus_{j\in W} w_j]\,.$$

Proof We first evaluate the probability $\Pr[f(w) = \bigoplus_{j\in S} w_j]$:

$$\begin{aligned}
\Pr[f(w) &= \bigoplus_{j\in W} w_j] = \frac{1}{2^n}\sum_w [\bigoplus_{j\in W} w_j f(w) + (1 - \bigoplus_{j\in W} w_j)(1 - f(w))] \\
&= \frac{1}{2^n}\sum_w [1 - \bigoplus_{j\in W} w_j - f(w) + 2\bigoplus_{j\in W} w_j f(w)] \\
&= 1 - \frac{1}{2^n}\sum_w \bigoplus_{j\in W} w_j - \frac{1}{2^n}\sum_w f(w) + \frac{2}{2^n}\sum_w \bigoplus_{j\in W} w_j f(w) \\
&= 1 - \frac{1}{2} - p + \frac{2}{2^n}\sum_w \bigoplus_{j\in W} w_j f(w) \\
&= \frac{1}{2} - \left(p - \frac{2}{2^n}\sum_w \bigoplus_{j\in W} w_j f(w)\right).
\end{aligned}$$

By substituting $(-1)^{\sum_{j\in W} w_j} = 1 - 2\left(\bigoplus_{j\in W} w_j\right)$ in the definition of the coefficient $f^*(W)$, we then obtain

$$\begin{aligned}
f^*(W) &= \frac{1}{2^n}\sum_w (-1)^{\sum_{j\in W} w_j} f(w) \\
&= \frac{1}{2^n}\sum (1 - 2\bigoplus_{j\in W} w_j) f(w) \\
&= \frac{1}{2^n}\sum_f f(w) - \frac{2}{2^n}\sum_w \bigoplus_{j\in W} w_j f(w) \\
&= p - \frac{2}{2^n}\sum_w \bigoplus_{j\in W} w_j f(w),
\end{aligned}$$

and the thesis follows.

□

There is no correlation if $f^*(W) = 0$, and maximum correlation if $|f^*(W)| = \frac{1}{2}$. The sign of the coefficient indicates if the correlation is actually with the parity ($f^*(W) = -\frac{1}{2}$) or with its complement ($f^*(W) = \frac{1}{2}$).
Notice that the first order coefficients $f^*(\{i\})$, $1 \le i \le n$, measure the correlation of the function f with its i-th variable:

$$f^*(\{i\}) = \frac{1}{2} - \Pr[f(w) = w_i].$$

2.2 Sensitivity of Boolean functions

The *sensitivity* of a Boolean function is a measure of how the value of the function is sensitive to changes in the input.

Definition 2.5 *Let f be a Boolean function depending on n binary variables.*

- *The* sensitivity *of f on the string $w \in \{0,1\}^n$, $s_f(w)$, is the number of locations i for which $f(w) \neq f(w^{(i)})$, i.e.*

$$s_f(w) = \sum_{i=1}^{n} (f(w) - f(w^{(i)}))^2 \,.$$

- *The* average sensitivity *of f, $s(f)$, is the average of $s_f(w)$ over all $w \in \{0,1\}^n$, taken with respect to the uniform distribution:*

$$s(f) = \frac{1}{2^n} \sum_{w} s_f(w) = \mathbf{E}[s_f(w)] \,.$$

- *The* maximal sensitivity *of f is defined as*

$$S_{max}(f) = \max_{w} \{s_f(w)\} \,.$$

(Sometimes we will use the terminology *sensitivity of a set* as a shortcut for *sensitivity of the characteristic function of the elements of length n of a set.*)

This measure has been discussed in the literature under various names, e.g. it is sometimes called 'critical complexity' (see [73]).

Simon [62] showed that every function that depends on all its variables has maximal sensitivity of at least $\Omega(\log n)$. This bound can be improved in the case of symmetric functions:

Proposition 2.2 *For any symmetric and nonconstant function depending on n binary variables*

$$S_{max}(f) \geq \frac{n+1}{2} \,.$$

Proof Let $f^{(k)}$ denote the value taken by f on input strings of cardinality k, $0 \leq k \leq n$, and $s^{(k)}$ the value of the sensitivity of f on such strings. Since f is nonconstant, it must exist an integer k such that $f^{(k)} \neq f^{(k+1)}$. Then, we get

$$\begin{aligned} s^{(k)} &\geq n-k \\ s^{(k+1)} &\geq k+1 \,, \end{aligned}$$

and the thesis immediately follows:

$$S_{max}(f) \geq \max\{n-k, k+1\} \geq \frac{n+1}{2} \,.$$

□

We now focus our attention on the average sensitivity and show that it turns out to be a *submodular formal measure of complexity* (see [59] and [73]). First of all, we recall the definition of *formal complexity measure* and of *submodular formal complexity measure.*

Definition 2.6 ([59]) *A nonnegative real-valued function μ defined on the set of all Boolean functions in n variables is a* formal complexity measure *if*

1. $\mu(x_i) \leq 1, \qquad \mu(\neg x_i) \ \leq \ 1, \qquad (1 \leq i \leq n)$;
2. $\mu(f \vee g) \leq \mu(f) + \mu(g) \qquad$ for each $f,\ g$;
3. $\mu(f \wedge g) \leq \mu(f) + \mu(g) \qquad$ for each $f,\ g$.

Definition 2.7 ([59]) *A nonnegative real-valued function μ defined on the set of all Boolean functions in n variables is a* submodular formal complexity measure *if it satisfies the condition 1 and the following* submodularity condition

4. $\mu(f \vee g) \ + \ \mu(f \wedge g) \ \leq \ \mu(f) + \mu(g) \qquad$ for each $f,\ g$,

which is stronger than both 2 and 3.

Lemma 2.2 *The average sensitivity is a submodular measure of complexity.*

Proof In the following proof we will make use of the notion of influences of variables on a function (see Section 2.4 for the definition).

Conditions *1* and *4* are easily seen to be satisfied if one considers the definition of the average sensitivity of a Boolean function as sum of the influences of its variables. Let us consider the functions $f_i(x_1, x_2, \ldots, x_n) = x_i$, for $1 \leq i \leq n$. Since for $1 \leq i, j \leq n$ $I_{f_i}(x_j) = \delta_{ij}$, we get $s(f_i) = 1$ and condition *1* is satisfied.

Now, let f and g be two Boolean functions defined on $\{0,1\}^n$. Let us consider the functions $(f \vee g)(x_1, x_2, \ldots, x_n)$ and $(f \wedge g)(x_1, x_2, \ldots, x_n)$. By using some probabilistic arguments related to the definition of influence, we obtain

$$I_{f \vee g}(x_i) + I_{f \wedge g}(x_i) \leq I_f(x_i) + I_g(x_i), \qquad 1 \leq i \leq n\,.$$

Then, from the fact that the average sensitivity can be defined as sum of the influences of the variables, we get

$$s(f \vee g) \ + \ s(f \wedge g) \ \leq \ s(f) \ + \ s(g)\,,$$

and condition *4* is satisfied.

2.2.1 Distribution of the Average Sensitivity

We now analyze the distribution of the average sensitivity of all Boolean functions defined on $\{0,1\}^n$ and give an exact evaluation of its expected value and variance.

First of all, note that, for any function f, $0 \leq s(f) \leq n$ and that $s(f)$ can take only rational values.

Theorem 2.2 *The average sensitivity of all 2^{2^n} Boolean functions of n variables has expected value $\frac{n}{2}$ and variance $\frac{n}{2^{n+2}}$.*

Proof The first claim easily follows from the linearity of the expected value. In fact:

$$\begin{aligned}
E[s(f)] &= E\left[\frac{1}{2^n}\sum_w s_f(w)\right] = \frac{1}{2^n}\sum_w E[s_f(w)] \\
&= \frac{1}{2^n}\sum_w \sum_{|u|=1} E[f(w)+f(w\oplus u)-2f(w)f(w\oplus u)] \\
&= \frac{1}{2^n}\sum_w \sum_{|u|=1} (E[f(w)]+E[f(w\oplus u)]-2E[f(w)f(w\oplus u)]) \\
&= \frac{1}{2^n}\sum_w \sum_{|u|=1} \left(\frac{1}{2}+\frac{1}{2}-2\cdot\frac{1}{4}\right) = \frac{n}{2}.
\end{aligned}$$

For the second claim, from the definition of variance we get

$$V[s(f)] = E[s^2(f)] - E^2[s(f)].$$

Then, we must evaluate the expected value of $s^2(f)$. First of all, we have

$$\begin{aligned}
E[s^2(f)] &= E\left[\frac{1}{2^{2n}}\sum_w \sum_{w'} s_f(w)s_f(w')\right] \\
&= \frac{1}{2^{2n}}E\left[\sum_w s_f^2(w) + \sum_{w\neq w'} s_f(w)s_f(w')\right] \\
&= \frac{1}{2^{2n}}\left(\sum_w E[s_f^2(w)] + \sum_{w\neq w'} E[s_f(w)s_f(w')]\right).
\end{aligned}$$

Since

$$E[s_f^2(w)] = \sum_{I=1}^{n} E[d_i^2(w)] + \sum_{i\neq j} E[d_i(w)d_j(w)],$$

where $d_i(w) = f(w)+f(w\oplus i)-2f(w)f(w\oplus i)$, and i is a string with the i-th bit equal to 1 and the others equal to 0, by multiplying $d_i(w)$ and $d_j(w)$ and evaluating the expected values, we obtain

$$E[s_f^2(w)] = \frac{n^2}{4} + \frac{n}{4}.$$

In the same way, we find

$$E[s_f(w)s_f(w')] = \sum_{i,j} E[d_i(w)d_j(w')] = \frac{n^2}{4}.$$

Note that $E[d_i(w)d_j(w')]$ turns out to be equal to $1/4$ even if w and w' share a Hamming neighbor, i.e. $w\oplus i = w'\oplus j$ for some i and j, and even if w is a Hamming neighbor of w', i.e. $w = w'\oplus j$ for some j (and *vice-versa*). Finally, we get

$$E[s^2(f)] = \frac{1}{2^{2n}}\left[2^n\left(\frac{n^2}{4}+\frac{n}{4}\right) + \left(2^{2n}-2^n\right)\frac{n^2}{4}\right] = \frac{n}{2^{n+2}} + \frac{n^2}{4},$$

and the thesis follows from $E[s(f)] = \frac{n}{2}$.

$\square$

Lemma 2.3 *The distribution of the average sensitivity of all the 2^{2^n} Boolean functions of n variables is symmetric with respect to the expectation, i.e.*

$$\#\{\ f : \{0,1\}^n \to \{0,1\}\ \mid\ s(f) = k\ \}| = \#\{\ f : \{0,1\}^n \to \{0,1\}\ \mid\ s(f) = n - k\ \}|\,,$$

Proof We show that for any Boolean function f, with average sensitivity $s(f)$, there exists a function g s.t. $s(g) = n - s(f)$. We define g in the following way. For all $w \in \{0,1\}^n$,

$$g(w) = \begin{cases} 1 - f(w) & \text{if } |w| \text{ is odd}\,, \\ f(w) & \text{if } |w| \text{ is even}\,. \end{cases}$$

The function g is such that, for all w, $s_g(w) = n - s_f(w)$. In fact for all the strings u such that $|u| = 1$, w and $w \oplus u$ have opposite parities, therefore we obtain

$$\begin{aligned} s_g(w) &= \sum_{|u|=1} |g(w) - g(w \oplus u)| \\ &= \sum_{|u|=1} (1 - |f(w) - f(w \oplus u)|) = n - s_f(w)\,. \end{aligned}$$

Hence

$$s(g) = \frac{1}{2^n} \sum_w s_g(w) = n - s(f)\,.$$

$\square$

Note that from the two constant functions $f(w) = 0$ and $g(w) = 1 - f(w) = 1$ for all w, which are the only two functions with average sensitivity equal to 0, we can use Theorem 2.2 to get the two functions with the maximum average sensitivity, i.e., the parity and its complement.

2.3 Harmonic Analysis and Sensitivity

We now give an interpretation of the harmonic analysis on the hypercube in terms of its *Laplacian Matrix*, and will use it below, to describe the interplay between the Fourier coefficients and the notion of sensitivity.

First of all, recall that the *n-dimensional hypercube* consists of 2^n vertices $\{0,1\}^n$, and that there is an edge between two vertices if and only if they differ in exactly one coordinate. Since each vertex has n adjacent edges, the n-dimensional hypercube is an n-regular graph. We now define its *adjacency* and *Laplacian* matrices.

Definition 2.8

- *The* $2^n \times 2^n$ adjacency matrix *of the* n*-dimensional hypercube is recursively defined as:*

$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad A_n = \begin{pmatrix} A_{n-1} & I_{n-1} \\ I_{n-1} & A_{n-1} \end{pmatrix},$$

where I_k *denotes the* $2^k \times 2^k$ *identity matrix.*

- *The* $2^n \times 2^n$ Laplacian matrix *of the* n*-dimensional hypercube is defined as:*

$$\mathcal{L}_n = nI_n - A_n \, .$$

The following theorem shows the intimate relationship between the Laplacian matrix $\mathcal{L}_n$ and the Hadamard matrix H_n. For any integer $0 \le i \le 2^n - 1$, we denote by $|i|$ the number of ones in its binary expansion $b(i) \in \{0,1\}^n$.

Theorem 2.3 *For any* n, $\mathcal{L}_n = \frac{1}{2^{n-1}} H_n D_n H_n$ *, where* H_n *is the Hadamard matrix and* D_n *is the* $2^n \times 2^n$ *diagonal matrix whose* (i,i)*-entry is equal to* $|i|$*, for* $0 \le i \le 2^n - 1$*.*

Proof By induction on n.

Base

By direct inspection.

Induction step

We consider the $2^{n+1} \times 2^{n+1}$ Laplacian matrix $\mathcal{L}_{n+1}$. Since

$$\mathcal{L}_{n+1} = \begin{pmatrix} \mathcal{L}_n + I_n & -I_n \\ -I_n & \mathcal{L}_n + I \end{pmatrix} \quad \text{and} \quad D_{n+1} = \begin{pmatrix} D_n & 0 \\ 0 & D_n + I_n \end{pmatrix},$$

the theorem immediately follows by induction hypothesis, i.e.,

$$\begin{aligned} \frac{1}{2^n} H_{n+1} D_{n+1} H_{n+1} &= \frac{1}{2^n} \begin{pmatrix} 2H_n D_n H_n + H_n H_n & -H_n H_n \\ -H_n H_n & 2H_n D_n H_n + H_n H_n \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{L}_n + I_n & -I_n \\ -I_n & \mathcal{L}_n + I_n \end{pmatrix}. \end{aligned}$$

□

Corollary 2.1 *H_n is the matrix of the eigenvectors of the Laplacian matrix (as well as of the adjacency matrix) of the n-dimensional hypercube.*

Proof Easily follows from Theorem 2.3, since $H_n^{-1} = \frac{1}{2^n} H_n$.

$\square$

The eigenvalues of the Laplacian matrix are given by $\lambda(i) = 2|i|$, for any $0 \leq i \leq 2^n - 1$.

Lemma 2.4 shows how it is possible to express the sensitivity of Boolean functions in terms of the Laplacian matrix.

Let f be a Boolean function depending on n variables and $\tilde{s}$ the vector defined as $\tilde{s} = \mathcal{L}_n f$.

Lemma 2.4 *For any $0 \leq i \leq 2^n - 1$, the absolute value of the i^{th} component of $\tilde{s}$ is equal to the sensitivity of f on the string $b(i)$; more precisely:*

$$\tilde{s}_i = -(-1)^{f(b(i))} \; s_f(b(i)) \, .$$

Proof Let us consider the i^{th} entry of the vector $\tilde{s} = \mathcal{L}_n f$. We have

$$\tilde{s}_i = nf(b(i)) - (A_n f)_i = nf(b(i)) - \sum_{|\mu|=1} f(b(i) \oplus \mu) \, .$$

Thus we obtain

$$\tilde{s}_i = \begin{cases} n - \sum_{|\mu|=1} f(b(i) \oplus \mu) = s_f(b(i)) & \text{if } \; f(b(i)) = 1 \, , \\ - \sum_{|\mu|=1} f(b(i) \oplus \mu) = -s_f(b(i)) & \text{if } \; f(b(i)) = 0 \, . \end{cases}$$

$\square$

Corollary 2.2 *For a Boolean function f, we have:*

(i) $S_{max}(f) = \|\mathcal{L}_n f\|_\infty$.

(ii) $s(f) = \frac{1}{2^n} \|\mathcal{L}_n f\|_1$.

Proof Immediate from Lemma 2.4.

$\square$

We now use these results to analyze the connections between sensitivity and Fourier coefficients. We establish a relation between the maximal sensitivity and the Fourier coefficients of a function, and between the average sensitivity and the Fourier spectrum. The latter was actually already proved, with a different and more complicated proof, in the now largely unknown [36].

Lemma 2.5

1. $S_{max}(f) = \max_w \left| 2 \; \sum_v (-1)^{v^T w} |v| f^*(v) \right|$.

2. $s(f) = \frac{1}{2^{n-1}} f^T \mathcal{L}_n f = 4 \sum_w |w| (f^*(w))^2$.

Proof

1. From Theorem 2.3 and from the definition of Fourier transform of a Boolean function, it follows that

$$\tilde{s} = \mathcal{L}_n f = \frac{2}{2^n} H_n D_n H_n f = 2 H_n D_n f^* ,$$

which yields

$$|\tilde{s}_i| = 2 \left| \sum_v (-1)^{b(i)^T v} |v| f^*(v) \right| .$$

The proof thus follows from the equality $|\tilde{s}_i| = s_f(b(i))$.

2. Let u denote the vector whose entries are all equal to 1. Lemma 2.4 yields

$$s(f) = \frac{1}{2^n} (2f - u)^T \mathcal{L}_n f = \frac{1}{2^{n-1}} f^T \mathcal{L}_n f ,$$

where the last equality follows since $u^T \mathcal{L}_n f = 0$, as one can easily check. Then, from Theorem 2.3 we can derive

$$\begin{aligned} s(f) &= \frac{1}{2^{n-1}} f^T \mathcal{L}_n f = \frac{1}{2^{2(n-1)}} f^T H_n D_n H_n f \\ &= 4 (f^*)^T D_n (f^*) = 4 \sum_w |w| (f^*(w))^2 . \end{aligned}$$

□

2.4 Fourier Coefficients and Influences of Sets of variables

In this section we analyze the links between harmonic analysis and the notion of *influences* of sets of variables. Let $\mathcal{A}$ be a set of variables.

Definition 2.9 *The* influence of $\mathcal{A}$ on f, *denoted by $I_f(\mathcal{A})$, is the probability that f remains undetermined as long as the variables in $\mathcal{A}$ are not assigned values and the other variables are assigned at random according to the uniform distribution.*

Notice that the average sensitivity of a function can be equivalently defined as the sum of the influences of all variables on f (see [38]):

$$s(f) = \sum_{i=1}^{n} I_f(x_i) .$$

The relations between influences and Fourier coefficients are clear when the function is monotone and only for first order coefficients. It was observed in [38] that for any monotone function f, and any variable x_i

$$I_f(x_i) = 2\ |f^*(\{i\})|\,.$$

This also holds for *locally monotone* Boolean function (see [34]), i.e. functions that are monotonic increasing or decreasing in each variable. On the other hand, for general Boolean functions we get

$$I_f(x_i) \geq 2\ |f^*(\{i\})|\,.$$

In [34] it was also proved that for any function, and any variable x_i

$$I_f(x_i) \geq 2 \max_{\{I:i\in I\}} \{|f^*(I)|\}\,.$$

From this result, it follows that for locally monotone functions, the linear Fourier coefficients are the largest in absolute value.

We find exact relationships between influences of subsets of variables and Fourier coefficients of any order. For the sake of exposition we present first the case of first order coefficients and then the general case.

First order coefficients. Let $w = x_1x_2\ldots x_n \in \{0,1\}^n$ and let $\tilde{w}$ denote the string obtained from w by replacing the i-th bit x_i with an undefined value $\star$, i.e. $\tilde{w} = x_1x_2\ldots x_{i-1}\star x_{i+1}\ldots x_n \in \{0,1\}^n$. Let $S_i \subseteq \{0,1\}^n$ be the set

$$S_i \equiv \{w \in \{0,1\}^n \quad | \quad x_i = \star \quad \Rightarrow \quad f(\tilde{w}) = \star\}.$$

Let $S_i^{\oplus} \equiv \{w \in S_i \quad | \quad f(w) = x_i\}$ and $S_i^{\overline{\oplus}} \equiv \{w \in S_i \quad | \quad f(w) = \overline{x}_i\}$. Note that $S_i = S_i^{\oplus} \cup S_i^{\overline{\oplus}}$ and $S_i^{\oplus} \cap S_i^{\overline{\oplus}} = \emptyset$.

Definition 2.10 *The* influence of the variable x_i on the function f *is given by*

$$I_f(x_i) = \frac{1}{2^n}\ |S_i| = \frac{1}{2^n}\sum_w [1 - f(w)f(w\oplus i) - \overline{f(w)}\ \overline{f(w\oplus i)}]\,.$$

Definition 2.11 *The* positive influence *and the* negative influence *of the variable x_i on f are given by*

$$\begin{aligned} I_f^{\oplus}(x_i) &= \frac{1}{2^n}|S_i^{\oplus}| = \frac{1}{2^n}\sum_{w\in S_i}(x_i f(w) + \overline{x}_i\ \overline{f(w)})\,,\\ I_f^{\overline{\oplus}}(x_i) &= \frac{1}{2^n}|S_i^{\overline{\oplus}}| = \frac{1}{2^n}\sum_{w\in S_i}(x_i\overline{f(w)} + \overline{x}_i f(w))\,. \end{aligned}$$

Observe that, for any $1 \leq i \leq n$, we have $I_f(x_i) = I_f^{\oplus}(x_i) + I_f^{\overline{\oplus}}(x_i)$.

Proposition 2.3 *For any* $1 \le i \le n$, $f^*(\{i\}) = \frac{1}{2}$ [$I_f^{\overline{\oplus}}(x_i) - I_f^{\oplus}(x_i)$].

The known relation between influences and the first order coefficients of locally monotone functions easily follows from Proposition 2.3. In fact, for any locally monotone function and any variable x_i, one of the two components of the influence must vanish, and thus we obtain

$$|f^*(\{i\})| = \frac{1}{2}\ I_f(x_i)\,.$$

For the special case of monotone functions, we get $I_f^{\overline{\oplus}}(x_i) = 0$, from which

$$f^*(\{i\}) = -\frac{1}{2}\ I_f^{\oplus}(x_i) = -\frac{1}{2}\ I_f(x_i)\,.$$

General case. Let $S_{i_1,\ldots,i_k} \subseteq \{0,1\}^n$ be a set defined as

$$S_{i_1,\ldots,i_k} \equiv \{w = x_1x_2\ldots x_n \in \{0,1\}^n \mid x_{i_1} = \star, \ldots, x_{i_k} = \star \ \Rightarrow\ f(\tilde{w}) = \star\}\,,$$

where $\tilde{w}$ is the string obtained from w by replacing the bits $x_{i_1}, \ldots, x_{i_k}$ with undefined values $\star$. Then, let $S^{\oplus}_{i_1,\ldots,i_k}$ and $S^{\overline{\oplus}}_{i_1,\ldots,i_k}$ be the following sets

$$S^{\oplus}_{i_1,\ldots,i_k} \equiv \left\{ w = x_1x_2\ldots x_n \in S_{i_1,\ldots,i_k} \quad | \quad f(w) = \bigoplus_{l=i_1}^{i_k} x_l \right\}\,,$$

$$S^{\overline{\oplus}}_{i_1,\ldots,i_k} \equiv \left\{ w = x_1x_2\ldots x_n \in S_{i_1,\ldots,i_k} \quad | \quad f(w) = \overline{\bigoplus_{l=i_1}^{i_k} x_l} \right\}.$$

Note that $S_{i_1,\ldots,i_k} = S^{\oplus}_{i_1,\ldots,i_k} \cup S^{\overline{\oplus}}_{i_1,\ldots,i_k}$ and $S^{\oplus}_{i} \cap S^{\overline{\oplus}}_{i_1,\ldots,i_k} = \emptyset$.

Definition 2.12 *The* influence of the variables $x_{i_1}, \ldots, x_{i_k}$ on the function f *is given by*

$$I_f(x_{i_1}, \ldots, x_{i_k}) = \frac{1}{2^n}\ |S_{i_1,\ldots,i_k}|\,.$$

Definition 2.13 *The* positive influence, $I_f^{\oplus}$*, and the* negative influence, $I_f^{\overline{\oplus}}$*, of the variables* $x_{i_1}, \ldots, x_{i_k}$ *on* f *are given by*

$$\begin{aligned} I_f^{\oplus}(x_{i_1}, \ldots, x_{i_k}) &= \frac{1}{2^n}|S^{\oplus}_{i_1,\ldots,i_k}| = \frac{1}{2^n} \sum_{w \in S_{i_1,\ldots,i_k}} \left(\bigoplus_{l=i_1}^{i_k} x_l\ f(w) + \overline{\bigoplus_{l=i_1}^{i_k} x_l}\ \overline{f(w)} \right)\,, \\ I_f^{\overline{\oplus}}(x_{i_1}, \ldots, x_{i_k}) &= \frac{1}{2^n}|S^{\overline{\oplus}}_{i_1,\ldots,i_k}| = \frac{1}{2^n} \sum_{w \in S_{i_1,\ldots,i_k}} \left(\bigoplus_{l=i_1}^{i_k} x_l\ \overline{f(w)} + \overline{\bigoplus_{l=i_1}^{i_k} x_l}\ f(w) \right)\,. \end{aligned}$$

For all subsets of k variables $\{x_{i_1}, \ldots, x_{i_k}\}$, we get

$$I_f(x_{i_1}, \ldots, x_{i_k}) = I_f^{\oplus}(x_{i_1}, \ldots, x_{i_k}) + I_f^{\overline{\oplus}}(x_{i_1}, \ldots, x_{i_k}) .$$

Proposition 2.4 *For all subsets of k variables $\{x_{i_1}, \ldots, x_{i_k}\}$,*

$$f^*(\{i_1, \ldots, i_k\}) = \frac{1}{2} \; [\; I_f^{\overline{\oplus}}(x_{i_1}, \ldots, x_{i_k}) - I_f^{\oplus}(x_{i_1}, \ldots, x_{i_k}) \;].$$

Note that these results yield the relation $I_f(A) \geq 2 \; |f^*(A)|$, for any Boolean function and any subset $A \subseteq \{1, 2, \ldots, n\}$.

2.5 Fourier Analysis of Symmetric and Monotone functions

A basic computational problem involving Fourier coefficients is the following. Given a Boolean function f - say by the circuit C that computes it, obtain the Fourier coefficients of f - say by exhibiting an algorithm with input a set S (given by its characteristic vector) that outputs the corresponding Fourier coefficient of f, as a quotient of two integers. It is easy to see that the value of these integers is at most exponential, so they can be written in time polynomial in n. It is also clear from the previous paragraphs that the computation takes at most exponential time. It is an interesting open problem to find classes of Boolean functions for which we can compute the Fourier coefficients in time polynomial in the size of the circuits.

We know of three such classes: *random* functions (i.e. almost all Boolean functions), *symmetric* and *monotone* Boolean functions. It is unlikely that a general algorithm exists, since it would imply $P = PP$.

The existence of an algorithm for random functions is trivial: the coefficients can be computed in exponential time, which is polynomial in the size of the circuit (which is $\frac{2^n}{n}$).

As we have seen, the coefficient f_0^* is proportional to the number of assignments that give f the value 1, so computing this coefficient in P would yield a polynomial time algorithm for a PP-complete problem. On the other hand, it is not hard to see that every coefficient can be computed in polynomial time with a PP oracle.

We sketch below how to compute the Fourier coefficients and estimate the sensitivity of symmetric functions and monotone functions. We also state some properties of the Fourier coefficients of random Boolean functions.

Symmetric functions

The Fourier coefficients of symmetric functions can be expressed in terms of the Krawtchouk polynomials $K_k(x; n)$, $x \in R$, defined by

$$K_k(x; n) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j} .$$

First of all, note that given a binary string w of length n, we get

$$f^*(w) = 2^{-n} \sum_{k=0}^{n} f_{2^k-1} \sum_{|x|=k} (-1)^{w^T x} .$$

Then, we observe that

$$\sum_{|x|=k} (-1)^{w^T x} = \sum_{i=0}^{k} (-1)^i \begin{pmatrix} n-|w| \\ k-i \end{pmatrix} \begin{pmatrix} |w| \\ i \end{pmatrix} ,$$

and finally we obtain

$$f^*(w) = 2^{-n} \sum_{k=0}^{n} f_{2^k-1} K_k(|w|; n) .$$

For the average sensitivity we get

$$s(f) = 4n \sum_{i=1}^{n} \begin{pmatrix} n-1 \\ i-1 \end{pmatrix} \left(f^*_{2^i-1}\right)^2 .$$

Monotone functions

The n coefficients which are sufficient to determine the sensitivity [38] can be computed as $f^* = \hat{H}_n f$, where $\hat{H}_n$ is an $n \times 2^n$ matrix defined as

$$\hat{H}_1 = \begin{pmatrix} 1 & -1 \end{pmatrix} , \qquad \hat{H}_k = \begin{pmatrix} \hat{H}_{k-1} & \hat{H}_{k-1} \\ v_{k-1} & -v_{k-1} \end{pmatrix} ,$$

where $v_i = (1, 1, \ldots, 1)^T$. Then $s(f) = 2^{-n+1} \|\hat{H}_n f\|_1$.

Threshold functions

In the case of monotone and symmetric functions, i.e. *thresholds*, we get

$$f^*(w) = 2^{-n} \sum_{k=h}^{n} K_k(|w|; n) = -2^{-n} K_{h-1}(|w|-1; n-1) ,$$

where h is the positive integer which defines the threshold, and where we used the following property of Krawtchouk polynomials

$$\sum_{k=0}^{l} K_k(x; n) = K_l(x-1; n-1) .$$

In addition, note that for every w', $w \in \{0,1\}^n$ such that $|w'| = n - |w| + 1$ we have $|f^*(w)| = |f^*(w')|$.

Finally, the average sensitivity of h-threshold functions is given by

$$s(f) = \frac{h}{2^{n-1}} \begin{pmatrix} n \\ h \end{pmatrix} .$$

An important threshold function is majority. For this function, which we denote by $\textsc{Maj}$, we have the following results. For n odd,

$$\textsc{Maj}^*(w) = \begin{cases} \frac{1}{2} & \text{if} \quad |w| \quad \text{is} \quad 0 \\ (-1)^{\frac{|w|+1}{2}} \frac{1}{2^n} \frac{\binom{|w|-1}{\frac{|w|-1}{2}}\binom{n-|w|}{\frac{n-|w|}{2}}}{\binom{\frac{n-1}{2}}{\frac{|w|-1}{2}}} & \text{if} \quad |w| \quad \text{is} \quad \text{odd} \\ 0 & \text{if} \quad |w| \quad \text{is} \quad \text{even.} \end{cases}$$

Moreover, using Stirling's approximation, we obtain, for $|w|$ odd,

$$|\textsc{Maj}^*(w)| \approx \frac{1}{\sqrt{\pi}} \left[\frac{(|w|-1)^{|w|-1}(n-|w|)^{n-|w|}}{(n-1)^n} \right]^{\frac{1}{2}} .$$

Lemma 2.6 *Let δ be a positive constant such that $\frac{\sqrt{n}}{k\sqrt{k(n-k)}} \leq \frac{1}{k^{1+\delta}}$. Then*

$$\sum_{t<|S|\leq n-t} (\textsc{Maj}^*\{S\})^2 = \sum_{k=t}^{n-t} E_k(\textsc{Maj}) \leq \frac{1}{\pi\sqrt{2\pi}} \left[\frac{1}{\delta\ t^\delta} + o\left(\frac{1}{t^\delta}\right) \right] .$$

Proof The result follows from the fact that

$$\pi\sqrt{2\pi} \sum_{t<|S|\leq n-t} (\textsc{Maj}^*\{S\})^2 \leq \sum_{k=t}^{n-t} \frac{1}{k^{1+\delta}} \leq \frac{1}{t^{1+\delta}} + \int_t^{n-t} \frac{1}{x^{1+\delta}} dx \leq \frac{1}{t^{1+\delta}} + \frac{1}{\delta}\frac{1}{t^\delta}.$$

□

Random functions

We now state some properties of the Fourier coefficients of random Boolean functions. We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is a random Boolean function if, for all $w \in \{0,1\}^n$, we have $\Pr\{f(w) = 0\} = \Pr\{f(w) = 1\} = \frac{1}{2}$. The next lemma gives the characterization of the Fourier coefficients of a random function.

Lemma 2.7 *Let f be a random Boolean function. Then*

(a) *The expected value of each, but the first, Fourier coefficient is equal to 0.*

(b) *The variance of each Fourier coefficient is equal to* $\frac{1}{2^{n+2}}$.

Proof

(a) Note that $E[f(w)] = \frac{1}{2}$. Then, the thesis easily follows from the matrix formulation of the Fourier transform $f^* = 2^{-n}H_n f$, by exploiting the structural properties of the Hadamard symmetric transform matrix H_n and the linearity of the expected value. With respect to the first Fourier coefficient, which represents the probability that f takes the value 1, we obviously obtain $E[f_0^*] = \frac{1}{2}$.

(b) From the definition of variance, and from the fact that $E[f_j^*] = \frac{1}{2}\delta_{0j}$, for $0 \leq j \leq 2^n - 1$, we get $V[f_j^*] = E[(f_j^*)^2] - \frac{1}{4}\delta_{0j}$. Then, the thesis follows by evaluating $E[(f_j^*)^2]$. We can work as in part **(a)**, by exploiting the properties of H_n and the linearity of the expected value.

□

Note that, from the value of the variance and from the fact that it must result $\sum_{j=0}^{2^n-1}(f_j^*)^2 \sim \frac{1}{2}$, it follows that the Fourier coefficients of a random function are indeed "small", but they vanish with low probability. More precisely, one can verify, by looking at the structure of the Hadamard matrix, that the probability for a coefficient to be zero is of the order of $\binom{2^{n-1}}{2^{n-2}}^2\binom{2^n}{2^{n-1}}^{-1} \sim \frac{1}{\sqrt{\pi 2^{n-3}}}$. Also note that these results agree with the interpretation of the Fourier coefficients as measures of the correlation between the function and the parity of subsets of the input bits (see Section 2.1).

2.6 Generalized Sensitivity

The average sensitivity is a measure of how a function behaves when one input bit is flipped. In [13], we have considered a generalization of the average sensitivity by allowing a few - rather than just one - bits to be flipped. This may be useful in order to distinguish functions with high average sensitivity but with an overall very regular behavior, like parity, from functions which are hard to compute.

Definition 2.14 *Let f be a Boolean function defined on $\{0,1\}^n$. The k-*sensitivity *$s_f^{(k)}(w)$ of f on a string $w \in \{0,1\}^n$ is the number of strings $\hat{w}$ such that $d(w,\hat{w}) = k$ and $f(w) \neq f(\hat{w})$, i.e.*

$$s_f^{(k)}(w) = \sum_{|u|=k} (f(w) - f(w \oplus u))^2 .$$

*The k-*sensitivity *of f, $s^{(k)}(f)$, is the average of $s_f^{(k)}(w)$ over all $w \in \{0,1\}^n$:*

$$s^{(k)}(f) = \frac{1}{2^n}\sum_w s_f^{(k)}(w) .$$

Finally, the maximal k-sensitivity, $S^{(k)}_{max}(f)$, *is the maximum of* $s^{(k)}_f(w)$ *over all* $w \in \{0,1\}^n$.

Note that the 1-sensitivity coincides with the average sensitivity. Moreover, as for the average sensitivity, the k-sensitivity can take only rational values.

Like the average sensitivity, the generalized sensitivity has a strong relation with the Fourier spectrum of a Boolean function.

Lemma 2.8 *Let f be a Boolean function defined on $\{0,1\}^n$, and let f^* be the vector of its Fourier coefficients. Then*

$$s^{(k)}(f) = 4\sum_w W_k(|w|)\ (f^*(w))^2,$$

where the weight function $W_k(|w|)$ *is defined as follows*

$$W_k(|w|) = \sum_{l=0}^{\lceil \frac{k}{2} \rceil -1} \binom{|w|}{2l+1} \binom{n-|w|}{k-(2l+1)}.$$

Proof We take advantage of Parseval's identity and of the autocorrelation identity (see Section 2.1). We obtain, for the k-sensitivity $s^{(k)}_f(w)$ of a string w,

$$s^{(k)}_f(w) = \sum_{|u|=k} (f(w) - f(w \oplus u))^2 = \sum_{|u|=k} (f(w) + f(w \oplus u) - 2f(w)f(w \oplus u)),$$

from which

$$s^{(k)}_f(w) = \binom{n}{k} f(w) + \sum_{|u|=k} f(w \oplus u) - 2\sum_{|u|=k} f(w)f(w \oplus u).$$

For the k-sensitivity of the function f, some algebra yields

$$s^{(k)}(f) = 2^{-n+1} \left[\binom{n}{k} \sum_w f(w) - \sum_{|u|=k} \sum_w f(w)f(w \oplus u) \right].$$

Since we have

$$\sum_{|u|=k} \sum_w f(w)f(w \oplus u) = 2^n \binom{n}{k} \sum_w (f^*(w))^2 - 2^{n+1} \sum_w W_k(|w|)\ (f^*(w))^2,$$

then, by Parseval's equality, we obtain

$$s^{(k)}(f) = 4\sum_w W_k(|w|)\ (f^*(w))^2.$$

□

The explicit expression of the k-sensitivity in terms of the Fourier coefficients, for $1 \le k \le 4$, is given by:

$$\begin{aligned}
s^{(1)}(f) &= 4 \sum_w |w| \; (f^*(w))^2 \\
s^{(2)}(f) &= 4 \sum_w |w| \; (n - |w|) \; (f^*(w))^2 \\
s^{(3)}(f) &= 4 \sum_w \left[|w| \binom{n - |w|}{2} + \binom{|w|}{3} \right] (f^*(w))^2 \\
s^{(4)}(f) &= 4 \sum_w \left[|w| \binom{n - |w|}{3} + (n - |w|) \binom{|w|}{3} \right] (f^*(w))^2 .
\end{aligned}$$

With respect to the notion of formal complexity measure (see Section 2.2), one can easily verify that, since the k-sensitivity of the function $f(x_1, x_2, \ldots, x_n) = x_i$ is equal to the binomial coefficient $\binom{n-1}{k-1}$, then condition *1* in Definition 2.6 is not satisfied if $k \ge 2$. In order to obtain a measure of complexity, we can normalize $s^{(k)}(f)$ as follows:

$$\hat{s}^{(k)}(f) = \binom{n-1}{k-1}^{-1} s^{(k)}(f) .$$

Lemma 2.9 *The normalized k-sensitivity, $\hat{s}^{(k)}(f)$, is a submodular measure of complexity.*

Proof Condition *1* in Definition 2.6 is obviously satisfied. Let f and g be two Boolean functions defined on $\{0,1\}^n$. For all $w \in \{0,1\}^n$ we get

$$\hat{s}_f^{(k)}(w) + \hat{s}_g^{(k)}(w) = \sum_{|u|=k} \left[(f(w) - f(w \oplus u))^2 + (g(w) - g(w \oplus u))^2 \right] .$$

Now, let us consider the functions $(f \vee g)(w) = f(w) \vee g(w)$ and $(f \wedge g)(w) = f(w) \wedge g(w)$. For all $w \in \{0,1\}^n$ we have

$$\begin{aligned}
\hat{s}_{f \vee g}^{(k)}(w) &= \sum_{|u|=k} (f(w) \vee g(w) - f(w \oplus u) \vee g(w \oplus u))^2 , \\
\hat{s}_{f \wedge g}^{(k)}(w) &= \sum_{|u|=k} (f(w) \wedge g(w) - f(w \oplus u) \wedge g(w \oplus u))^2 .
\end{aligned}$$

From these expressions is easy to verify, e.g. by considering all the sixteen possible combinations of the value of $f(w)$, $g(w)$, $f(w \oplus u)$ and $g(w \oplus u)$, that

$$\hat{s}_{f \wedge g}^{(k)}(w) + \hat{s}_{f \vee g}^{(k)}(w) \le \hat{s}_f^{(k)}(w) + \hat{s}_g^{(k)}(w) .$$

Then, by averaging over all w, the submodularity condition of Definition 2.7 is easily seen to be satisfied.

□

We now focus our attention on the 2-sensitivity. First of all, notice that the maximum value achievable by $s^{(2)}(f)$ does not coincide with the maximum value achievable by $S^{(2)}_{max}(f)$, i.e. $\binom{n}{2}$. This follows from neighborhood properties of strings in the n-cube, i.e. intersections between strings at Hamming distance 2. In general these intersections take place whenever we consider even Hamming distances. For this reason, for even k, we have $0 \leq s^{(k)}(f) < \binom{n}{k}$. For odd k, we have $0 \leq s^{(k)}(f) \leq \binom{n}{k}$ and the maximum value $\binom{n}{k}$ is achieved by the parity function and its complement. With respect to the 2-sensitivity, we have that $\frac{n^2}{4} \leq \max_f\{s^{(2)}(f)\} < \binom{n}{2}$, where the value $\frac{n^2}{4}$ is achieved by the parity function computed on $\frac{n}{2}$ input bits (for simplicity, we have assumed that n is even; when n is odd, we can consider the parity function computed on $\frac{n-1}{2}$ input bits, which has 2-sensitivity equal to $\frac{n^2-1}{4}$).

We now derive some additional properties of the 2-sensitivity of symmetric Boolean functions.

Let w be a binary string of length n, and let $\hat{w}$ denote its k-neighbors. For even $k = 2h$, there are $\binom{n-|w|}{h}\binom{|w|}{h}$ k-neighbors $\hat{w}$ such that $|\hat{w}| = |w|$. Therefore, for any symmetric Boolean function f, we have $s^{(2h)}(f) \leq \binom{n}{2h} - \frac{1}{2^n}\sum_w \binom{n-|w|}{h}\binom{|w|}{h}$. On the contrary, for odd k, there exist symmetric functions with k-sensitivity of the order of $\binom{n}{k}$, e.g. the parity function.

By using this result, we can prove the following Lemma.

Lemma 2.10 *The 2-sensitivity $s^{(2)}(f)$ cannot achieve its maximum value over the set of the symmetric Boolean functions.*

Proof First of all, we have that for any symmetric function

$$s^{(2)}(f) \leq \binom{n}{2} - \frac{1}{2^n}\sum_w (n-|w|)\,|w| = \frac{1}{2}\binom{n}{2}.$$

Then, the thesis follows from the existence of nonsymmetric Boolean functions with 2-sensitivity greater than $\frac{1}{2}\binom{n}{2}$, e.g. the parity function computed on $\lfloor \frac{n}{2} \rfloor$ input bits. $\square$

Note that the bound given by the previous Lemma is tight, in fact the value $\frac{1}{2}\binom{n}{2}$ can be detected by looking at functions with spectra of the form 00110011...0011.

2.7 The Spectral Lemma

We now present an interesting application of harmonic analysis to Circuit Complexity, due to Linial et al. (see [45]). They studied how the computational complexity of Boolean functions is related to their Fourier transform. Specifically, they studied the Fourier transform of functions computable by constant depth circuits and derived an inequality that the transform of such functions satisfies. This inequality was then used to establish new results on complexity and learnability of constant depth circuits.

As we have already mentioned, one of the best known results in circuit complexity is that AC^0 circuits require exponential size to compute the parity function. More precisely, AC^0 circuits cannot even approximate the parity function. This fact has a direct consequence on the Fourier transform, because, as we have already mentioned, the Fourier coefficients measure the correlation between the function and the parity of subsets of its input bits. Consequently, each high order Fourier coefficient of an AC^0 function must be very small (where "high order" means coefficients corresponding to strings of large cardinality). By exploiting this fact, Linial et al. were able to prove that not only is each individual high order coefficient small, but in fact the sum of squares (i.e. the *power spectrum*) associated with all high Fourier coefficients is very small.

Lemma 2.11 [**Spectral Lemma** [45]] *Let f be a Boolean function on n variables computable by a Boolean circuit of depth d and size M, and let t be any integer. Then*

$$\sum_{|w|>t} (f^*(w))^2 \leq \frac{1}{2}\, M\, 2^{-\frac{t^{1/d}}{20}} .$$

The main tool of the proof of the Spectral lemma is the well known *Håstad's Switching Lemma* [35], which states that AC^0 functions tend to simplify in a significant way when subjected to random restrictions (see Section 1.4 for the definitions of restrictions and random restrictions). Precisely, Linial et al. used a stronger version of the Switching lemma, together with a deep analysis of how the Fourier transform of f relates to the transform of its random restrictions, to develop the necessary machinery to prove the Spectral lemma (see [45] for details of the proof).

Since the Switching lemma represents one of the most important and powerful tools to prove lower bounds on small depth circuits, let us spend a few words on it.

The Switching lemma. Several of the lower bound proofs on the size of small depth circuits have the same outline. Let us consider the case of the parity function. The proofs that the parity does not belong to the class AC^0 ([33], [35], and [75]) are by induction and proceed as follows:

1. prove that the parity circuits of depth 2 are large;

2. prove that small depth k parity circuits can be converted to small depth $k-1$ parity circuits.

Of these two steps, the first one is easy, while the second step is much more difficult and here lies the difference between the various proofs. The Switching lemma provides exactly a tool to carry through this outline of the lower bounds proofs. It can be stated, in an informal way, as follows. Let a *small gate* denote a gate with few inputs.

Lemma 2.12 *Given a depth two circuit which is an AND of small OR gates, then if random values are substituted for a randomly selected subset of the variables, it is possible to write the resulting induced function as an OR of small AND gates with very high probability.*

Using this lemma, lower bounds are easily proved by induction over the depth. Indeed, if we hit a depth k circuit with a random restriction, then we can change an AND of ORs to an OR of ANDs without increasing the size. In this way, we obtain a circuit with two adjacent levels consisting of OR gates, which can be merged to one level, giving a depth $k-1$ circuits (this fails with only exponentially small probability). Formally, we have the following statement:

Lemma 2.13 [**Håstad's Switching Lemma** [35]] *Let f be an AND of ORs all of size at most t, and choose a random restriction ρ with parameter p (i.e., $\Pr[\rho(x_i) = \star] = p$). Then, the probability that f_ρ cannot be written as an OR of ANDs all of size less than s is bounded by α^s, where α is the unique positive root to the equation*

$$\left(1 + \frac{4p}{\alpha(1+p)}\right)^t = \left(1 + \frac{2p}{\alpha(1+p)}\right)^t + 1\,.$$

An upper bound on the size of α is provided by $5pt$, for $p < p_0$ for some absolute constant $p_0 > 0$ (see [35]).

Consequences of the Spectral Lemma. The Spectral lemma implies several new properties of functions in AC^0.

One of the most interesting applications is an $O(n^{polylog(n)})$-time algorithm for learning functions in AC^0. The algorithm proceeds as follows: it first observes the behavior of an AC^0 function on $O(n^{polylog(n)})$ randomly chosen inputs; this allows it to derive (with high probability) a very good approximation to all the low order Fourier coefficients of the function. Since the high order coefficients are guaranteed by the Spectral lemma to be very small, these approximations of the low order coefficients are informative enough to predict the behavior of the function on inputs chosen uniformly at random. Moreover, since there are only a few low order coefficients, the approximation can be done efficiently (see [45] for more details). Notice that, as we already pointed out, in recent years other results in Learning Theory have been obtained by applying the Fourier-based techniques.

Another consequence of the Spectral lemma is that every function in AC^0 can be approximated well by a real polynomial of low (*polylog*) degree. This complements the results of [58] and [64], showing that such an approximation is possible over finite fields.

Finally, an interesting application of the Spectral lemma provides an upper bound on the average sensitivity of AC^0 functions. Indeed, Lemma 2.11, together with Lemma 2.5, implies:

Lemma 2.14 ([45]) *For any function $f \in AC^0[d]$, we have $s(f) = O((\log n)^d)$.*

Thus, every function in AC^0 has low average sensitivity to its input. Changing one bit of the input is very unlikely to change the value of the function, when the original input and the bit are chosen at random. Note that the bound given by this lemma is not far from optimal as the parity function on $(\log n)^{d-1}$ bits has sensitivity $(\log n)^{d-1}$, and can be computed in $AC^0[d]$.

Lemma 2.14 and, more in general, Lemma 2.11, give a general, simple way to prove lower bounds for AC^0. For example, we have used them to prove that strongly 'balanced' functions cannot belong to AC^0 (see Chapter 4). Moreover, we have been able to improve the bound of Lemma 2.14 in the case of symmetric AC^0 functions (see Section 2.8.1).

2.8 Applications to Boolean Function Complexity

In this section, we present some applications of Fourier Analysis to Circuit Complexity. More interesting applications will be presented in Chapter 4.

2.8.1 Symmetric AC^0 functions

The upper bound on the average sensitivity of AC^0 functions can be dramatically improved in the case of symmetric AC^0 functions.

In this section we first show that there is a correspondence between sparsity or co-sparsity and functions whose average sensitivity decreases exponentially. This fact is the basis for showing that low average sensitivity is a structural property of sets which generalizes sparsity in a natural way. Then we find some interesting relations between a measure of complexity for symmetric functions defined in [31] and the average sensitivity. These relationships allow us to use a result of [25] for proving that the average sensitivity of symmetric functions - and, more in general, of functions with polynomial index - in AC^0 decreases exponentially.

Definition 2.15 *We say that a language over the alphabet $\{0, 1\}$ is*

(i) generalized sparse *if, for any n, the number of strings of length n which belong to it is either at most $n^{O(1)}$ or at least $2^n - n^{O(1)}$;*

(ii) generalized almost sparse *if the number of strings of length n which belong to it is either at most $n^{polylogn}$ or at least $2^n - n^{polylogn}$.*

Note that if we have a generalized sparse language L (for which we know, for any length n, if it has at most $n^{O(1)}$ or at least $2^n - n^{O(1)}$ elements) then SAT cannot be reduced to L (assuming $P \neq NP$). Indeed there is a language L' that L (and therefore, SAT) is 1-truth-table reducible to, and by [54], this implies $P = NP$.

Lemma 2.15 *A set is generalized sparse if and only if the average sensitivity of its characteristic function is $O\left(\frac{n^{O(1)}}{2^n}\right)$. A set is generalized almost sparse if and only if the average sensitivity of its characteristic function is $O\left(\frac{n^{polylogn}}{2^n}\right)$.*

Proof From the definition of average sensitivity, we have that $s(f) \le 2n\ min\{p,q\}$, where $p = Pr\{f(w)=1\}$, $q = 1-p$, and the probability is taken over all $w \in \{0,1\}^n$. The *if* part of the Lemma follows from the fact that the characteristic function of a generalized sparse set satisfies either $p \le \frac{n^{O(1)}}{2^n}$ or $q \le \frac{n^{O(1)}}{2^n}$. It remains to prove that the functions with exponentially low average sensitivity are the characteristic functions of generalized sparse sets. From Parseval's identity and from the definition of f_0^*, we get

$$p = \sum_i (f_i^*)^2 = f_0^{*2} + \sum_{i \neq 0} (f_i^*)^2 \le f_0^{*2} + \sum_{i\neq 0} |i| (f_i^*)^2 = f_0^{*2} + \frac{s(f)}{4},$$

i.e. $p^2 - p + \frac{s(f)}{4} \ge 0$. Solving the latter inequality for p and using the hypothesis that $s(f) \le \frac{n^{O(1)}}{2^n}$ we obtain that either $p \le \frac{n^{O(1)}}{2^n}$ or $q \le \frac{n^{O(1)}}{2^n}$, i.e. f is the characteristic function of a generalized sparse set. The proof for generalized almost sparse sets is similar.

□

Let f be a symmetric Boolean function. We recall some definitions from [31]. The minimum number of variables of f that have to be set to constant values so that f becomes a constant function is called *measure* of f and is denoted by $\mu(f)$. Let $\sigma \in \{0,1\}^{n+1}$ with elements σ_i, where σ_i is equal to the value of f when i variables are set to 1 and the other variables are set to 0. σ is called *spectrum* of f. σ_i is called i-th *character* of σ. A *subword* of the spectrum is a string of the form $\sigma_i\, \sigma_{i+1} \ldots \sigma_{i+k}$. [31] show that $\mu(f)$ can be easily evaluated from the spectrum because $\mu(f) = n+1-\Gamma$, where Γ is the length of the longest constant subword of σ.

Lemma 2.16 *Let f be a symmetric Boolean function defined on $\{0,1\}^n$ with measure $\mu(f)$. Then*

$$\frac{n}{2^{n-1}} \sum_{k=0}^{\bar{k}} \left\{ \binom{n-1}{\lfloor\frac{\mu(f)}{2}\rfloor - k\Gamma - 1} + \binom{n-1}{\lceil\frac{\mu(f)}{2}\rceil - k\Gamma - 1} \right\} \le s(f) \le \frac{n}{2^{n-1}} \sum_{k=0}^{\mu(f)-1} \binom{n-1}{k},$$

where $\bar{k} = \lfloor \frac{\mu(f)}{2\Gamma} \rfloor$. Furthermore, both the lower and the upper bounds are tight.

Proof The influence of x_i on a symmetric function f is given by

$$I_f(x_i) = \frac{1}{2^{n-1}} \sum_{k=0}^{n-1} \binom{n-1}{k} |\sigma_{k+1} - \sigma_k|,$$

where σ_k denotes the k-th character of the spectrum of f. Hence we have

$$s(f) = \frac{n}{2^{n-1}} \sum_{k=0}^{n-1} \binom{n-1}{k} |\sigma_{k+1} - \sigma_k|.$$

We need evaluate the maximum value which can be attained by the average sensitivity for functions whose measure is $\mu(f)$. By analyzing the behavior of the binomial coefficients $\binom{n-1}{k}$ for $0 \leq k \leq n-1$, it is easy to see that this maximum can be detected by looking at spectra of one of the forms

$$\sigma = \underbrace{1010101\ldots10}_{\mu}\underbrace{11111\ldots1}_{\Gamma};$$

$$\sigma = \underbrace{11111\ldots1}_{\Gamma}\underbrace{0101010\ldots10}_{\mu}$$

$$\sigma = \underbrace{1010101\ldots01}_{\mu}\underbrace{00000\ldots0}_{\Gamma};$$

$$\sigma = \underbrace{00000\ldots0}_{\Gamma}\underbrace{1010101\ldots01}_{\mu}$$

Thus, the maximum average sensitivity of functions with measure $\mu(f)$ is

$$\frac{n}{2^{n-1}}\sum_{k=0}^{\mu(f)-1}\binom{n-1}{k}.$$

The lower bound on $s(f)$ can be evaluated similarly. In fact, the functions of measure $\mu(f)$ whose average sensitivity is minimal must have spectra of the form:

$$\underbrace{000\ldots0}_{\lfloor\frac{\mu(f)}{2}\rfloor-\bar{k}\Gamma}\underbrace{\underbrace{111\ldots1}_{\Gamma}\underbrace{000\ldots0}_{\Gamma}\ldots\underbrace{111\ldots1}_{\Gamma}}_{\bar{k}\Gamma}\underbrace{000\ldots0}_{\Gamma}\underbrace{\underbrace{111\ldots1}_{\Gamma}\ldots\underbrace{000\ldots0}_{\Gamma}\underbrace{111\ldots1}_{\Gamma}}_{\bar{k}\Gamma}\underbrace{000\ldots0}_{\lceil\frac{\mu(f)}{2}\rceil-\bar{k}\Gamma},$$

where $\bar{k} = \lfloor\frac{\mu(f)}{2\Gamma}\rfloor$. Hence there exist functions of average sensitivity

$$\frac{n}{2^{n-1}}\sum_{k=0}^{\bar{k}}\left\{\binom{n-1}{\lfloor\frac{\mu(f)}{2}\rfloor-k\Gamma-1}+\binom{n-1}{\lceil\frac{\mu(f)}{2}\rceil-k\Gamma-1}\right\},$$

which is the minimum possible value for functions of measure μ.

□

The above Lemma has a very interesting consequence for symmetric functions in AC^0.

Theorem 2.4 *Let f be a symmetric function in AC^0. Then $s(f) = O(2^{-n+polylog\,n})$, and this is equivalent to saying that f is the characteristic function of a generalized almost sparse language.*

Proof A consequence of Lemma 2.16, together with the characterization given by [31], is that symmetric functions in AC^0 have exponentially decreasing average sensitivity. In fact, since symmetric functions in AC^0 have measure bounded above by a polylog, from Lemma 2.16 we obtain

$$s(f) \leq \frac{n}{2^{n-1}}\sum_{k=0}^{polylogn}\binom{n-1}{k} \leq \frac{n^{polylogn}}{2^n}.$$

Then Lemma 2.15 implies that f is the characteristic function of a generalized almost sparse language.

$\square$

Corollary 2.3 $\theta\left(n^{polylogn}\right)$ *symmetric functions of* n *variables are computable by polynomial size constant depth circuits.*

Proof The upper bound follows from Theorem 1 in [74] and standard counting arguments. The lower bound is obtained by counting the number of functions for which $\mu(f) = O(polylog\, n)$.

$\square$

These results show that symmetric functions in AC^0 are such that the influence of variables must be almost exponentially small. The same is clearly not true for nonsymmetric functions. Thus if one wants to look for interesting functions in AC^0, she should concentrate on nonsymmetric functions.

2.8.2 Composition of Boolean functions

We characterize the Fourier transform of the Boolean function composition $h = f(g_1, g_2, \ldots, g_m)$, in terms of the Fourier coefficients of f and of the g_i. We hope that this will eventually enable us to prove lower bound theorems using a 'top-down' approach. For example, we would like to prove that a circuit with a mod6 gate at the top, and AC^0 functions as inputs to the MOD_6 gate cannot compute majority, by using properties of the spectra of the MOD_6 function, of AC^0 functions, and of the composition matrix. Unfortunately, we are unable to prove such a theorem yet: our estimates are not yet precise enough, and the formulas, although explicitly computable, are complicated.

Throughout this section a Boolean function will be defined as $f : \{1,-1\}^n \to \{1,-1\}$; namely, 0 and 1 are represented by 1 and -1, respectively. Note that this changes the Fourier spectrum only by a constant factor:

$$f^*_{\{0,1\}}(w) = \begin{cases} -\frac{1}{2} f^*_{\{1,-1\}}(w) & \text{if} \quad |w| \neq 0 \\ \frac{1}{2}(1 - f^*_{\{1,-1\}}(w)) & \text{if} \quad |w| = 0\,, \end{cases}$$

where $f^*_{\{0,1\}}$ and $f^*_{\{1,-1\}}$ are the coefficients of the function considered over $\{0,1\}$ and $\{1,-1\}$ respectively.

Let $f : \{1,-1\}^m \to \{1,-1\}$ and $g_i : \{1,-1\}^n \to \{1,-1\}$, for $i = 1, \ldots, m$. Consider the Boolean function $h : \{1,-1\}^n \to \{1,-1\}$ defined as

$$h(w) = f(g_1(w), g_2(w), \ldots, g_m(w)),$$

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

Let f^* and h^* be the vectors of the Fourier coefficients of the functions f and h, respectively. H_i denotes the Hadamard matrix of size 2^i. We say that $h_i \asymp f_j$ if the m-tuple $g_1(w)\, g_2(w) \ldots g_m(w)$, where $w = b(i)$, is equal to $b(j)$, and $b(i)$ and $b(j)$ denote the binary expansions of i and j, respectively.

Lemma 2.17 *Let f, g_i and h be as above and let V be a $2^n \times 2^m$ matrix defined as $2^{-n}\, H_n A H_m$, where A is a $2^n \times 2^m$ matrix whose entries α_{ij} satisfy*

$$\alpha_{ij} = \begin{cases} 1 & \text{if } h_i \asymp f_j \\ 0 & \text{otherwise} \end{cases}$$

Then $h^* = V f^*$.

Proof From the definition of the matrix A, we have $h = Af$. The thesis follows from the fact that $h^* = 2^{-n}\, H_n h$ and $f^* = 2^{-m}\, H_m f$.

□

Note that the columns of the matrix A are mutually orthogonal and thus the matrix $A^T A$ is a $2^m \times 2^m$ diagonal matrix whose (j,j)-th entry a_j is given by $\sum_{i=0}^{2^n-1} Pr\{h_i \asymp f_j\}$.

The matrix V contains all the information on the relationship between the Fourier coefficients of the functions f and h. The next lemma states an important property of V.

Lemma 2.18 $\|V\| = \sqrt{2^{m-n}\, \rho(A^T A)} = \sqrt{2^m\, p}$, *where $p = \max_j p_j$, and $p_j = 2^{-n} a_j$.*

Note that p (or a sharp estimate for it) can be easily computed for many functions. Since $2^{-m} \leq p \leq 1$, we have that $\|V\| \leq 2^{\frac{m}{2}}$, which shows that $2^{\frac{m}{2}}$ is the maximum amplification factor for the Fourier coefficients. In addition, note that, if we interpret the set of the functions g_i as a single mapping $\hat{g} : \{1,-1\}^n \rightarrow \{1,-1\}^m$, then a_j is the cardinality of the inverse image of the string $b(j)$ according to $\hat{g}$, i.e. $a_j = \#\{w \in \{1,-1\}^n \mid \hat{g}(w) = b(j)\}$.

It is possible to give a better characterization to the matrix V. We first need to consider the special case $f \equiv parity$.

Composition of functions with parity. Let $h : \{1,-1\}^n \rightarrow \{1,-1\}$ be defined as the parity of some others functions, i.e.

$$h(w) = g_1(w) \oplus g_2(w) \oplus \ldots \oplus g_m(w)\,,$$

where $w \in \{1,-1\}^n$. Since over the range $\{1,-1\}$ the parity function corresponds to a product, we have

$$h(w) = g_1(w) \cdot g_2(w) \cdot \ldots \cdot g_m(w).$$

This allows us to apply the *Convolution Theorem.*

Lemma 2.19 *The Fourier transform of the function h is given by the convolution of the Fourier spectra of the functions g_i, $i = 1, \ldots, m$:*

$$h^* = (g_1 \oplus g_2 \oplus \ldots \oplus g_k)^* = g_1^* * g_2^* * \ldots * g_m^* ,$$

where the convolution of two sequences $\{q(w)\}$ and $\{p(w)\}$ is defined as follows

$$(p * q)(w) = \sum_u p(u)q(w \oplus u).$$

Proof Let us first consider the case $m = 2$. Then,

$$\begin{aligned} h(w) &= g_1(w) \cdot g_2(w) \\ &= \sum_x (-1)^{w^T x} g_1^*(x) \cdot \sum_y (-1)^{w^T y} g_2^*(y) \\ &= \sum_x \sum_y (-1)^{w^T (x \oplus y)} g_1^*(x) g_2^*(y) . \end{aligned}$$

Thus, if we define $y = x \oplus c$, we obtain

$$\begin{aligned} h(w) &= \sum_x \sum_c (-1)^{w^T c} g_1^*(x) g_2^*(x \oplus c) \\ &= \sum_c (-1)^{w^T c} \sum_x g_1^*(x) g_2^*(x \oplus c) \\ &= \sum_c (-1)^{w^T c} (g_1^* * g_2^*)(c) . \end{aligned}$$

The case $m > 2$ easily follows since the convolution is an associative operation.

□

Observe that by the property of the convolution

$$\sum_w (p^*(w) * q^*(w)) = \sum_w p^*(w) \sum_w q^*(w) ,$$

it follows that $L_1(h) \leq L_1(g_1) \cdot L_1(g_2) \cdot \ldots \cdot L_1(g_m)$, where the L_1 norm associated with the spectrum of a function f is defined as $\sum_w |f^*(w)|$.

Characterization of the matrix V. We are now able to give a clear characterization to V. We first need some notation. Let $b(i)$ denotes the binary expansion of i, and $S(i) = \{j \mid \text{the } j^{th} \text{ bit in } b(i) \text{ is } 1\}$. Moreover, let v_i ($0 \leq i \leq 2^m - 1$) denotes the i^{th} column of the matrix V. Finally, let $\delta(w, u)$ be a function equal to 1 if $w = u$ and equal to 0 otherwise.

Lemma 2.20 *For any $0 \leq i \leq 2^m - 1$, v_i is given by the Fourier spectrum of the function*

$$p_j(w) = \bigoplus_{j \in S(i)} g_i(w) .$$

Proof For any $0 \leq i \leq 2^m - 1$, let $f(b_1, b_2, \ldots, b_m) = \bigoplus_{j \in S(i)} b_j$. Then $h(w) = \bigoplus_{j \in S(i)} g_j(w)$. The thesis then follows since $h^* = V f^*$ and since $f^*(w) = \delta_{w,b(i)}$ over the range $\{1, -1\}$.

$\square$

Corollary 2.4 *For any $0 \leq i \leq 2^m - 1$, the i^{th} column of V is given by the convolution of all the spectra g_j^*, with $j \in S(i)$.*

Thus, the Fourier spectrum of the function h is given by a linear combination of the Fourier spectra of the functions $p_i(w) = \bigoplus_{j \in S(i)} g_j(w)$, whose coefficients are the Fourier coefficients of f, i.e.

$$h^*(w) = \sum_i f^*(b(i)) p_i^*(w) \,.$$

Sensitivity: Composition and Amplification. We derive two upper bounds on the average sensitivity of h as a function of the average sensitivity of g_i and f, respectively.

Theorem 2.5 $s(h) \leq \sum_{i=1}^m s(g_i)$.

Proof (Sketch) Follows from the definition of sensitivity as a sum of influences of the variables and by using some probabilistic arguments on the propagation of influences from the functions g_i's to the function f.

$\square$

Theorem 2.6 $s(h) \leq n 2^{m-n} \rho(A^T A) s(f)$.

Proof Let D_n be a $2^n \times 2^n$ diagonal matrix whose (i,i)-th entry is defined as $|i|^{\frac{1}{2}}$. Then the average sensitivity of h can be written as

$$s(h) = \|D_n\, h^*\|^2 = \|D_n\, V\, f^*\|^2 \,.$$

Note that working with functions over the range $\{1, -1\}$, the factor 4 in the expression of the sensitivity in terms of Fourier coefficients, disappears. Now, if $\tilde{D}_m^{-1}$ is a $2^m \times 2^m$ diagonal matrix whose (i,i)-th entry is defined as $|i|^{-\frac{1}{2}}$, for $1 \leq i \leq 2^m$ and 0 for $i = 0$, we have that $\tilde{D}_m^{-1}\, D_m \,+\, e_1\, e_1^T \,=\, I$, from which

$$s(h) \,=\, \|D_n\, V\, (\, \tilde{D}_m^{-1}\, D_m \,+\, e_1\, e_1^T\,)\, f^*\|^2 \,,$$

where D_m is defined as D_n. Since $D_n\, V\, e_1 \;=\; 0$ and $s(f) \;=\; \|D_m\, f^*\|^2$, we get $s(h) \;\leq\; \|D_n\, V\, \tilde{D}_m^{-1}\|^2\, s(f)$, and the thesis now follows from Lemma 2.18 and from the equalities $\|D_n\,\|^2 \;=\; n$ and $\|\tilde{D}_m^{-1}\|^2 \;=\; 1$.

$\square$

The upper bound of Theorem 2.5 is a starting point for understanding the strict connection between parallel complexity and sensitivity. As as example, we have that, if the function f is the OR (AND) function and the functions g_i are AND (OR) functions, and if the fan-in of the $\wedge$ ($\vee$) gates is of order n, then $s(h) = O(c(n)\frac{n}{2^n})$, where $c(n)$ is the number of $\wedge$ ($\vee$) gates, from which $c(n) = \Omega(\frac{2^n}{n}s(h))$. (See also [20] for another bound on $c(n)$). Thus the notion of sensitivity 'explains' why some functions cannot be computed by circuits with polynomial size and small depth.

We consider now the upper bound of Theorem 2.6. There are two properties of composition which is worth analyzing, namely:

- Since the average sensitivity cannot exceed n and there are functions in NC^1 which attain this bound, for a study of completeness within NC^1 with respect to many-to-one reductions, it is interesting to find under which conditions we have $n2^{m-n}\rho(A^TA)s(f) < n$.

- Consider circuits over the basis $\{\wedge, \vee, \neg\}$. Since $s(AND) = s(OR) = \frac{n}{2^{n-1}}$ and since a polynomial number of AND/OR gates allows one to compute functions with sensitivity up to n, e.g. the parity of n bits, in general the amplification of the average sensitivity can have an exponential growth. Thus it is important to distinguish cases under which the composition of sensitivity has a polynomial, rather than exponential, amplification, i.e. $s(h) \leq n^{O(1)}s(f)$.

Let us interpret the set of the functions g_i as a single mapping $\hat{g} : \{1,-1\}^n \to \{1,-1\}^m$. We take into account two (extreme) cases.

- ***p* is large**. This corresponds to saying that the function $\hat{g}$ maps many strings $w \in \{1,-1\}^n$ onto a few strings $v \in \{1,-1\}^m$. In this case the upper bound is very large, but we can get direct information on h. In fact, if $p = 1$ then h is a constant function and thus $s(h) = 0$; if $p \geq 1 - [\alpha(n)]^{-1}$, where $\alpha(n)$ is any function increasing with n, then $s(h) = O(n[\alpha(n)]^{-1})$.

- ***p* is small**. This corresponds to saying that the cardinality of the inverse image of any string under $\hat{g}$ is approximately 2^{n-m}. If $2^{-m} \leq p \leq m^{O(1)}2^{-m}$, the we have $s(h) \leq n^{O(1)}s(f)$. If $2^{-m} \leq p \leq [\alpha(n)]^{-1}$, then

$$ns(f) \leq n2^{m-n}\rho(A^TA)s(f) \leq n[\alpha(n)]^{-1}2^m s(f)\,.$$

  If $p \simeq 2^{-m}$, then the functions f and h have approximately the same *density*, i.e. $2^{-m}\sum_w f(w) \simeq 2^{-n}\sum_v h(v)$, so that $\frac{s(f)}{n} \leq s(h) \leq ns(f)$.

As a consequence of these properties, we can use the upper bound of Theorem 2.6 to see that a sparse set cannot be complete in NC^1 under many-to-one reductions if $p = O(\frac{1}{\beta(m)})$, where $\beta(m)$ is any function growing more than any polynomial in m.

2.8.3 New Complexity Classes

We now define some new complexity classes, essentially composed of functions that have most of their power spectrum in a relatively narrow contiguous band. Such classes generalize the well-known class AC^0. We show that, like AC^0, they are learnable in quasi-polynomial time, under the uniform distribution. In fact, any class of Boolean functions with the property that most of its power spectrum is concentrated in a known set of coefficients, of at most polylogarithmic cardinality, is approximately learnable under the uniform distribution.

The symmetry which characterizes the average sensitivity distribution (see the proof of Lemma 2.3) suggests the definition of two complexity classes, which we call pAC^0 (for *parity* AC^0), and sAC^0 (for *symmetric* AC^0), where

$$\begin{aligned} pAC^0 &= \{g | g = f \oplus parity\,, \quad f \in AC^0\}\,, \\ sAC^0 &= AC^0 \cup pAC^0\,. \end{aligned}$$

These classes have the following properties.

- For any Boolean function $g \in pAC^0$ we have $s(g) = w(n - \log^k n)$, for a constant k, i.e. functions in pAC^0 behave similarly to parity or its complement (like functions in AC^0 behave similarly to the two constant functions).
- sAC^0 is a class which lies slightly above AC^0. In fact $sAC^0 \subset ACC$.
- We can use Theorems 2.2 and 2.8 in [44], to prove that, if $g = f \oplus parity$, then
$$g^*(w) = \frac{1}{2}\delta_{w,0} - \frac{1}{2}\delta_{w\oplus u,0} + f^*(w \oplus u)\,,$$
where u is the vector whose entries are all equal to 1, and $\delta_{i,j}$ is the Kronecker delta function. Thus the Fourier coefficients of order $|w|$ of g, $1 \le |w| < n$, coincide with the Fourier coefficients of order $n-|w|$ of f, while $g_0^* = \frac{1}{2} + f_{2^n-1}^*$, and $g_{2^n-1}^* = -\frac{1}{2} + f_0^*$.

This last symmetric property of the Fourier coefficients allows us to adapt results on the Fourier coefficients of AC^0 functions to pAC^0 functions. As an example, we have that functions in pAC^0 have almost all their power spectrum on the high order coefficients.

In general, it is interesting to compare the spectrum of a function f with that of a function g defined as $g(w) = f(w) \oplus p_m(w)$, where p_m is the parity of m bits, e.g. the first m bits. From the fact that $p_m(w) = w^T a (\text{mod } 2)$, where $a^T = [1, 1, \ldots, 1, 0, \ldots, 0]$, some algebra and the application of Theorem 2.8 in [44] yield

$$g^*(w_1, \ldots, w_n) = f^*(\bar{w_1}, \ldots, \bar{w_m}, w_{m+1}, \ldots, w_n)\,,$$

for $w \neq 0$ and $w \neq a$, and $g^*(0) = \frac{1}{2} + f^*(a)$, $g^*(a) = -\frac{1}{2} + f^*(0)$. Thus, all - but those for $w = 0$ and $w = a$ - the Fourier coefficients of f and g are the same, in different order. Some algebraic manipulation yields

$$s(g) = m - s(f) + 8 \sum_{w\neq 0, w\neq a} w^T(u - a)(f^*(w \oplus a))^2\,,$$

from which $s(g) \geq m - s(f)$.

The application of the above arguments to AC^0 functions gives rise to the definition of the complexity classes

$$\begin{aligned} p_i AC^0 &= \{g | g = f \oplus p_i, \quad f \in AC^0\} \\ cAC^0 &= \cup_{i=0}^{n} p_i AC^0 , \end{aligned}$$

where $p_0 AC^0 = AC^0$. We have

$$AC^0 \subset cAC^0 \subset ACC .$$

Generalizing further, we can define a class of Boolean functions that have at most k nonzero Fourier coefficients, or that have most of the power spectrum concentrated in such coefficients. It is not hard to see that every such class with $k = 2^{polylog}$ (as is the case for AC^0) is probably (approximately) learnable in time $2^{polylog}$ under uniform distribution. Simply sample the function at $2^{polylog}$ randomly chosen points, and determine the unknown Fourier coefficients solving the linear equations given by the Hadamard matrix H_n.

We now present an upper bound on the sum of the average sensitivity and the 2-sensitivity for the functions in the class sAC^0.

Theorem 2.7 *For any function $f \in sAC^0$ we have $[s + s^{(2)}](f) \leq n \ \log^{O(1)} n$.*

Proof For any function $f \in AC^0$ we have

$$[s + s^{(2)}](f) = (n+1) \ s(f) - 4 \ \sum_w |w| \ (f^*(w))^2 \ \leq \ n \ s(f).$$

Then the thesis follows from the fact that Boolean functions in AC^0 have average sensitivity $s(f) \leq \log^{O(1)} n$ [45].

Let us now consider a function g in pAC^0. Then $g = f \oplus parity$, with $f \in AC^0$, and $s(g) = n - s(f)$. Moreover, from the properties of the Fourier transform of functions in pAC^0, we obtain $s^{(2)}(g) = s^{(2)}(f)$, and the thesis easily follows.

Note that this result removes the differences in the behavior of the sensitivity of functions in AC^0 and pAC^0.

2.8.4 Formula Size Lower Bounds

In this section, we show how the Khrapchenko's theorem, which gives lower bounds on the size of Boolean formulas, can be restated in terms of Fourier coefficients. Following [41], let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function on n variables, let $\mathcal{L}(f)$ denote the number of leaves of the minimal-size $\{\vee, \wedge, \neg\}$-formula that computes f, and p the probability that f takes the value 1.

Lemma 2.21

$$\mathcal{L}(f) \geq \frac{1}{4p(1-p)} s(f)^2 = \frac{4}{p(1-p)} \left(\sum_w |w| (f^*(w))^2 \right)^2 .$$

Proof Let $A = f^{-1}(0)$ and $B = f^{-1}(1)$. Following [41], we define the $|A| \times |B|$ matrix Q, with $q_{ij} = 1$ if $i \in B$ and $j \in A$ differ in exactly one variable; otherwise, $q_{ij} = 0$. Khrapchenko's theorem can be restated as follows: $\mathcal{L}(f) \geq \frac{1}{|A| \cdot |B|} \left(\sum_{ij} q_{ij} \right)^2$. Then, the claim follows by noting that $\sum_{ij} q_{ij} = 2^{n-1} s(f) = 2^{n+1} \sum_w |w| (f^*(w))^2$, $|B| = 2^n p$ and $|A| = 2^n (1-p)$.

$\square$

A number of corollaries follows from this Lemma, for instance it is possible to prove that $\mathcal{L}(f) \geq \max_w |w|^2 (f^*(w))^4$.

Chapter 3

The Sensitivity versus Block Sensitivity Question

In [52] Nisan introduced the notion of *block sensitivity*, a generalization of the notion of maximal sensitivity, which gives a full characterization of the time needed to compute a Boolean function on a CREW PRAM with an unlimited number of processors. The block sensitivity is an extension of the maximal sensitivity, in which we are allowed to invert a block of bits simultaneously. This notion is different from the notion of k-sensitivity we introduced in Section 2.6, because we now consider disjoint blocks of, possibly, different size.

Because of the implications regarding spectral (i.e., mathematical) properties on the one side, and parallel complexity on the other side, it is very important to analyze the relationship between maximal sensitivity and block sensitivity of Boolean functions. It is obvious that the block sensitivity is at least as large as the maximal one, since the latter is just the block sensitivity with all blocks of length one. Rubinstein [60] exhibits an infinite family of Boolean functions with a quadratic gap between sensitivity and block sensitivity. This is the best known lower bound. Kenyon [40] proves upper bounds for the case where all blocks have the same size. It is a long-standing open problem whether the gap may be larger, e.g. superpolynomial.

As a starting point for attacking this intriguing combinatorial problem, we have considered *average* versions of these measures, and showed that for the family of functions considered by Rubinstein in [60], the average block sensitivity is $\Omega(\sqrt{n})$, while the average sensitivity is asymptotically zero. Thus, in the average case, there is no polynomial (or even exponential) relationship between maximal sensitivity and block sensitivity.

The chapter is organized as follows. In Section 3.1 we review some related work: we recall the definitions of *block sensitivity, certificate complexity* and *decision tree complexity* of Boolean functions, and examine their relationships (Section 3.1.1);

then, we briefly review some results linking the parallel time complexity of a Boolean function on a CREW PRAM to the previous measures of complexity (Section 3.1.2).

In Section 3.2 we consider the *average* version of the block sensitivity and discuss its relationship with the average sensitivity. Finally, in Section 3.3 we argue that the definition we give for the 'average' block sensitivity should be reasonable, since it preserves the difference between the sensitivity and the block sensitivity, while another possible definition yields exactly the average sensitivity. To prove this result we will make use of the Fourier based techniques.

Some of the results presented in this chapter appeared in [10].

3.1 Related Work

In this section we review some results concerning the relationships between several complexity measures of Boolean functions. The relationships we show will then have implications regarding CREW PRAM complexity (see [24], [26], [29], [43], [52], [72]).

3.1.1 Some Measures of Complexity for Boolean functions

The notion of *block sensitivity*, $BS(f)$, has been introduced by Nisan in [52], who used it to obtain a full characterization of the time needed to compute a Boolean function on a CREW PRAM with an unlimited number of processors (see Section 3.1.2). The block sensitivity is an extension of the maximal sensitivity, in which we are allowed to invert a block of bits simultaneously. Let $B \subseteq \{1, 2, \ldots, n\}$ be a subset of indices and let w be a Boolean string of length n. Then, $w^{(B)}$ means the string w with all bits in B flipped, i.e. $w^{(B)}$ differs from w exactly on B.

Definition 3.1 ([52]) *The function f is sensitive to the subset $B \subseteq \{1, 2, \ldots, n\}$ on w if $f(w) \neq f(w^{(B)})$. The* block sensitivity *of f on w, $BS_f(w)$, is the largest number t such that there exists t* disjoint *sets $B_1, B_2, \ldots, B_t$ such that, for all $1 \leq i \leq t$, f is sensitive to B_i on w. The* block sensitivity *of f, $BS(f)$, is the maximum over all w of the block sensitivity of f on w.*

It is obvious that for any function f, $S_{max}(f) \leq BS(f)$, since the sensitivity is just the block sensitivity with all blocks of length one.

We now review the notions of *deterministic decision tree complexity* and *certificate complexity.*

Definition 3.2 *The* deterministic decision tree complexity *of a function f, $D(f)$, is given by the minimal depth of a Boolean decision tree for n variables that computes f.*

Note that $D(f)$ is nothing but the number of bits read on the worst case input by the best deterministic algorithm in order to determine the value of the function f.

Definition 3.3 **([52])** *A* 1-certificate (0-certificate) *for f is an assignment to some subset of the variables that forces the value of f to 1 (0). The* certificate complexity of f on w, *$c_f(w)$, is the size of the smallest certificate that agrees with w. The* certificate complexity of f, *$C(f)$, is the maximum, over all w, of $c_f(w)$.*

The certificate complexity of a function describes how many bits of the input must be revealed to you (by someone who knows all the input bits) in order to convince you of the value of the function. This complexity measure has been widely used in the literature under various names; sometimes it is called 'sensitivity' ([72], [73]).

We now mention the known relationships between these complexity measures. We first consider the relations between maximal sensitivity, block sensitivity and certificate complexity.

Lemma 3.1 **([52])** *For any function f depending on n variables*

$$S_{max}(f) \leq BS(f) \leq C(f)\,.$$

Proof The left inequality follows directly from the definitions of maximal sensitivity and block sensitivity. The right inequality follows from the fact that for any input w, any certificate for w must include at least one variable from each set f is sensitive to on w.

□

It turns out that for a large subclass of functions these three measures of complexity are actually equal.

Proposition 3.1 **([52])** *For all monotone functions f*

$$S_{max}(f) = BS(f) = C(f)\,.$$

Proof It is enough to show that $C(f) \leq S_{max}(f)$. Consider a minimal certificate of size $C(f)$; without loss of generality assume it is a 1-certificate. The string which has 1 in every bit of the certificate and 0 in all other places, will have maximal sensitivity equal to $C(f)$. The reason is that whenever we flip any of the 1-bits, the value of the function will change from 1 to 0.

□

Notice that for general, nonmonotone, functions the inequalities can be strict. For instance, the symmetric function

$$f(w) = \begin{cases} 1 & \text{if } |w| = \frac{n}{2}\,, \frac{n}{2}+1\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

has maximal sensitivity $\frac{n}{2}+1$, block sensitivity $\frac{3}{4}\,n$, and certificate complexity $n-1$ (for simplicity, assume that 4 divides n).

Because of the implications regarding spectral (i.e., mathematical) properties and parallel complexity, it is important to study in a deeper way the relation between maximal and block sensitivity. In particular, it is important to compare how the maximal sensitivity and the block sensitivity behave on uniform families of functions.

Let $F \equiv \{f_n\}_{n \in I}$, for infinite $I \subseteq N$, denote an infinite uniform family of Boolean functions where, for any $n \in I$, $f_n : \{0,1\}^n \to \{0,1\}$. We will always drop the subscript and denote a fixed component of a family simply by f.

Given a family F, let $S_{max}(F)$, $BS(F) : N \to N$ denote the two functions that, on input n, are defined respectively as $S_{max}(f)$ and $BS(f)$, where f is the member of F depending on n variables. For families of monotone Boolean functions, we immediately get $S_{max}(F) \equiv BS(F)$. On the contrary, in the general case the relationship between these two functions is not clear. All we know is that $BS(F)$ and $S_{max}(F)$ are polynomially related when $BS(F) = \Omega(n)$ (see [40]). Moreover Rubinstein [60] exhibits an infinite family G of functions with a quadratic gap between the sensitivity and the block sensitivity, i.e.,

$$BS(G) = \Theta\left(S_{max}(G)^2\right) .$$

It is still an open problem whether the gap between the two measures may be bigger, e.g. super-polynomial. (Note that, in the case of symmetric functions, the gap can only be linear, since for any symmetric and nonconstant function f, $S_{max}(f) \geq \frac{n+1}{2}$, see Proposition 2.2.)

With respect to the relationship between block sensitivity and certificate complexity, Nisan proved that these two measure are polynomially related. In fact, there is the following lemma.

Lemma 3.2 **([52])** *For all Boolean functions f*

$$BS(f) \geq \sqrt{C(f)} .$$

Proof Let w be an input achieving the certificate complexity of f, i.e., every certificate for w is of length at least $C(f)$. Let B_1 be some minimal set of indices such that $f(w) \neq f(w^{(B_1)})$; let B_2 be another minimal set disjoint from B_1 and such that $f(w) \neq f(w^{(B_2)})$; and, in general, let B_i be a minimal set disjoint from all previous sets picked such that $f(w) \neq f(w^{(B_i)})$. We continue picking these sets until at a certain point no such set exists, and we call the last set B_t.

The union of all sets has to be a certificate for w, since otherwise we could have picked another set that changes the value of the function when flipped. Thus, we get

$$\sum_{i=1}^{t} |B_i| \geq C(f) .$$

We can now bound the block sensitivity of f in two ways:

(i) f is sensitive to each B_i on w, therefore $BS_f(w) \geq t$;

(ii) since for each i, B_i is minimal, then on $w^{(B_i)}$, f is sensitive to each element in B_i, thus $BS_f(w^{(B_i)}) \geq |B_i|$.

So, if $t \geq \sqrt{C(f)}$, then (i) gives us the desired result; otherwise at least one of the sets has to be of size larger than $\sqrt{C(f)}$ and (ii) will give us the result.

□

We conclude this section by mentioning the relationship between the decision tree complexity and the certificate complexity. This result was independently discovered by several people (see for instance [68]).

Lemma 3.3 **([68])** *For all Boolean functions f*

$$C(f) \leq D(f) \leq C^2(f)\,.$$

Finally, from Lemma 3.3 and Lemma 3.2 it follows that the decision tree complexity and the block sensitivity are polynomially related:

Corollary 3.1 *For all Boolean functions f*

$$BS(f) \leq D(f) \leq BS^4(f)\,.$$

3.1.2 Complexity of Boolean functions on CREW PRAMs

The PRAM (Parallel Random Access Machine) is a standard model for parallel computation. An ideal PRAM consists of an unbounded number of processors and an unbounded number of common memory cells that can be read and written by any processor. Each processor has its own local memory, and possibly unlimited computational power. Several variants of the PRAM model have been defined, which differ from each other in the way they handle memory access conflicts. Perhaps the most natural variant is the CREW PRAM, in which several processors may read from the same location at the same time, but two or more processors may never attempt to write into the same location at the same time.

Here we recall some results linking the parallel time complexity of a Boolean function on a CREW PRAM to its sensitivity, block sensitivity and certificate complexity.

Definition 3.4 *Let f be a Boolean function depending on n binary variables. The* CREW PRAM complexity *of f, $CREW(f)$, is the time needed to compute f on an ideal* CREW PRAM.

A key result bounding the power of ideal CREW PRAMs is the following theorem by Cook, Dwork and Reischuk [26], which provides a general lower bound for $CREW(f)$ in terms of the maximal sensitivity of f:

Theorem 3.1 **([26])** *For all Boolean functions f*

$$CREW(f) \geq \log_\alpha S_{max}(f)\,,$$

where α is some constant less than 5.

In [52], Nisan noticed that this result could be strengthened to give a bound in terms of the block sensitivity.

Lemma 3.4 **([52])** *For all Boolean functions f*

$$CREW(f) \geq \log_\alpha BS(f)\,.$$

The surprising fact is that Lemma 3.4 actually gives a tight lower bound for every function f. Indeed, by using the decision tree model of computation, Nisan first proved that for all Boolean functions f, $CREW(f) \leq \log D(f)$; then, since the decision tree complexity $D(f)$ and the block sensitivity $BS(f)$ are polynomially related (see Corollary 3.1), he obtained the following theorem:

Theorem 3.2 *For all Boolean functions f*

$$CREW(f) = \Theta\left(\log BS(f)\right)\,.$$

3.2 Average Sensitivity vs. Average Block Sensitivity

In this section we consider the *average* versions of sensitivity and block sensitivity and discuss their relationship.

As already mentioned, the average sensitivity is given by the average of $s_f(w)$ over all $w \in \{0,1\}^n$, taken with respect to the uniform distribution:

$$s(f) = \frac{1}{2^n}\sum_w s_f(w) = \mathbf{E}[s_f(w)]\,.$$

More care is needed in order to define the *average block sensitivity.* To this aim, we restate the definition of block sensitivity as follows.

Let $\mathcal{D}$ be the set of all the possible decompositions of the set $\{1,2,\ldots,n\}$ in disjoint blocks $B_1, B_2, \ldots, B_t$: $\mathcal{D} = \{\ \{B_i\}_{i\in I}\ |\ \ \cup_{i\in I} B_i = \{1,2,\ldots,n\},\ \ \forall\ i \neq j \in I\ \ B_i \cap B_j = \emptyset\ \}$. Let $d \in \mathcal{D}$ be a fixed decomposition $\{B_1, B_2, \ldots, B_k\}$.

Definition 3.5 *The* block sensitivity of f on w with respect to the decomposition *d is defined as:*

$$BS_{d,f}(w) = \sum_{i=1}^{k}\left(f(w) - f(w^{(B_i)})\right)^2\,.$$

The block sensitivity of f on w *is the maximum of* $BS_{d,f}(w)$ *over all the possible decompositions* d*:*

$$BS_f(w) = \max_{d \in \mathcal{D}} BS_{d,f}(w) \,.$$

Finally, the block sensitivity of f *is the maximum of* $BS_f(w)$ *over all* $w \in \{0,1\}^n$*:*

$$BS(f) = \max_{w} BS_f(w) = \max_{w} \; (\max_{d \in \mathcal{D}} BS_{d,f}(w)) \,.$$

We are now ready to define the *d-average block sensitivity* and the *average block sensitivity* of a function f.

Definition 3.6 *The* d-average block sensitivity *of a function* f*,* $bs_d(f)$*, is the average of* $BS_{d,f}(w)$ *over all* $w \in \{0,1\}^n$*, taken with respect to the uniform distribution:*

$$bs_d(f) = \mathbf{E}[BS_{d,f}(w)] \,.$$

The average block sensitivity *of a function* f*,* $bs(f)$*, is the average of* $BS_f(w)$ *over all* $w \in \{0,1\}^n$*, taken with respect to the uniform distribution:*

$$bs(f) = \mathbf{E}[BS_f(w)] = \mathbf{E}[\max_{d \in \mathcal{D}} BS_{d,f}(w)] \,.$$

Let us now consider the family $G \equiv \{g_n\}_{n \in I}$ of functions defined by Rubinstein (see [60]), where $I = \{\, (2m)^2 \mid m \in N \,\}$ and, for any $n \in I$, g_n is defined as

$$g(H) = \begin{cases} 1 & \text{if } H \cap \{i\sqrt{n}+1, \ldots, (i+1)\sqrt{n}\} = \{2j-1, 2j\} \\ & \quad \text{for some } i \text{ and } j,\;\; 0 \le i \le \sqrt{n}-1, \quad 1 \le j \le n/2 \\ 0 & \text{otherwise}\,, \end{cases}$$

where $H \subseteq \{1, 2, \ldots, n\}$. Note that we are using the set notation for g.

In [60] Rubinstein proved that $S_{max}(g) = \sqrt{n}$ and $BS(g) = \frac{n}{2}$. We show that, when we consider the average block sensitivity and the average sensitivity instead of the maximal ones, we have $bs(g) \ge \sqrt{n}$ and $s(g) \le t(n)$, where $\lim_{n \to \infty} t(n) = 0$.

Lemma 3.5 $bs(g) \ge \sqrt{n}\,.$

Proof We show that, for each $H \subseteq \{1, 2, \ldots, n\}$, $BS_g(H) \ge \sqrt{n}$. We consider different types of input.

- We first consider inputs H such that $g(H) = 1$ and $s_g(H) = \sqrt{n}$. These inputs are given by sets H with the following properties (see [60]):

  $$H \cap \{i\sqrt{n}+1, \ldots, (i+1)\sqrt{n}\} = \{2j-1, 2j\}$$

  for exactly one i, and for some j. In this case, we have $BS_g(H) \ge s_g(H) = \sqrt{n}\,.$

- We now consider inputs H such that $g(H) = 1$ and $s_g(H) < \sqrt{n}$. In this case, H must be of the type:

$$\underbrace{0\dots0110\dots0}_{\sqrt{n}}\Big|\underbrace{\cdots\cdots\cdots\cdots}_{\sqrt{n}}\Big|\underbrace{0\dots0110\dots0}_{\sqrt{n}}\Big|\underbrace{\cdots\cdots\cdots\cdots}_{\sqrt{n}}\Big|\underbrace{\cdots\cdots\cdots\cdots}_{\sqrt{n}}$$

  i.e. $H \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\} = \{2j-1, 2j\}$ for $k \geq 2$ different indices i, and some j, $1 \leq j \leq n/2$ (the case $k = 1$ has already been considered above).

  Let $\mathcal{L} = \{\ i \mid H \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\} = \{2j-1, 2j\}\ \}$. Then $|\mathcal{L}| = k \geq 2$.

  For these inputs, it is possible to define $\sqrt{n}$ blocks B_ℓ, of k bits each, such that by flipping all the bits whose index belongs to B_ℓ, we get, from H, a new set H_ℓ with the property that, $\forall\ i \in \mathcal{L}$:

$$H_\ell \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\} = \begin{cases} \{2j-1, 2j, p\} & \text{or} \\ \{2j-1\} & \text{or} \\ \{2j\}\,. \end{cases}$$

  In this way, we force the function to take the value 0. Then, since we can define $\sqrt{n}$ such sets B_ℓ, it follows that $BS_g(H) \geq \sqrt{n}\,.$

- Finally, we consider inputs H such that $g(H) = 0$. In this case, for each i and j

$$H \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\} \neq \{2j-1, 2j\}\,.$$

  Then, for each of the $\sqrt{n}$ sets $\{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\}$, it is possible to define at least one block B_i with the property that by flipping all the bits whose index belongs to B_i, we obtain from H a new set H_i such that $H_i \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\} = \{2j-1, 2j\}$ for some j. Then, we obtain $g(H_i) = 1$.

  Since we have $\sqrt{n}$ sets $\{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\}$, we get $BS_g(H) \geq \sqrt{n}\,.$

Thus, by putting these results together, we find $bs(g) = \mathbf{E}[BS_g(H)] \geq \sqrt{n}\,.$

□

Lemma 3.6 *$s(g) \leq t(n)$, where $\lim_{n\to\infty} t(n) = 0\,.$*

Proof First of all, let us evaluate the number of subsets $H \subseteq \{1, 2, \dots, n\}$ such that $s_g(H) = 0$. An easy lower bound on this quantity is given by the number of sets H such that $|H \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\}| = 0$ or $|H \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\}| \geq 4$, for any $0 \leq i \leq \sqrt{n}-1$. In fact, in this case, $g(H) = 0$ and there is no way to change this value to 1 by flipping only one bit. This is just a lower bound because there are also sets A of sensitivity 0 and such that $|A \cap \{i\sqrt{n}+1, \dots, (i+1)\sqrt{n}\}| = 2$ or 3 (e.g., when the bits equal to 1 are not adjacent in the string that corresponds to the set A). Then, since there are $\sqrt{n}$ bits in each set $\{i\sqrt{n}+1,\ldots,(i+1)\sqrt{n}\}$ and there are $\sqrt{n}$ such sets, we get

$$|\{H \mid s_g(H)=0\}| \geq \left[2^{\sqrt{n}} - \binom{\sqrt{n}}{1} - \binom{\sqrt{n}}{2} - \binom{\sqrt{n}}{3}\right]^{\sqrt{n}}.$$

By using the fact that $\binom{\sqrt{n}}{1}+\binom{\sqrt{n}}{2}+\binom{\sqrt{n}}{3} = \frac{\sqrt{n}(n+5)}{6} \leq n\sqrt{n}$, we obtain

$$|\{H \mid s_g(H)=0\}| \geq \left[2^{\sqrt{n}} - n\sqrt{n}\right]^{\sqrt{n}} = 2^n\left[1-\frac{n\sqrt{n}}{2^{\sqrt{n}}}\right]^{\sqrt{n}}.$$

Thus

$$|\{K \mid s_g(K)\neq 0\}| \leq 2^n - 2^n\left[1-\frac{n\sqrt{n}}{2^{\sqrt{n}}}\right]^{\sqrt{n}} = 2^n\left[1-\left(1-\frac{n\sqrt{n}}{2^{\sqrt{n}}}\right)^{\sqrt{n}}\right].$$

Since for each $K\subseteq\{1,2,\ldots,n\}$, $s_g(K)\leq\sqrt{n}$, for the average sensitivity we obtain

$$s(g) = \frac{1}{2^n}\sum_K s_g(K) \leq \frac{\sqrt{n}}{2^n}\,|\{K \mid s_g(K)\neq 0\}| \leq t(n),$$

where

$$t(n) = \sqrt{n}\left[1-\left(1-\frac{n\sqrt{n}}{2^{\sqrt{n}}}\right)^{\sqrt{n}}\right].$$

Finally, by using De L'Hopital's theorem it is possible to prove that $\lim_{n\to\infty} t(n) = 0$.
□

These results show that the family G considered by Rubinstein is such that $bs(G)=\Omega(\sqrt{n})$, while $s(G)$ is asymptotically zero. This means that, in the average case, there is no polynomial or even exponential relationship between the sensitivity and the block sensitivity of uniform families of Boolean functions.

3.3 Pseudo-average Block Sensitivity

We now consider a different definition of average block sensitivity, that we denote by $\widehat{\beta s}(f)$ and that we call *pseudo-average* block sensitivity. We show how this definition yields exactly the notion of average sensitivity. In the following analysis we will make use of Fourier based techniques.

Definition 3.7 *The* pseudo-average block sensitivity *of a Boolean function f is given by the maximum of the d-*average block sensitivity *over all the possible decompositions d:*

$$\widehat{\beta s}(f) = \max_{d\in\mathcal{D}} bs_d(f) = \max_{d\in\mathcal{D}} \mathbf{E}[BS_{d,f}(w)].$$

Lemma 3.7

(i) $\forall\, d \in \mathcal{D},\ \ bs_d(f) \leq s(f)$;

(ii) $\widehat{\beta s}(f) = s(f)$.

Proof We will prove (i) and (ii) by looking at the *spectral* definition of the average sensitivity. We first give an expression for $bs_d(f)$ in terms of the Fourier coefficients of f. In the derivation below we use Parseval's identity: $\sum_v f(v) = 2^n \sum_v (f^*(v))^2$, and the autocorrelation identity: $\sum_v f(v) f(v^{(U)}) = 2^n \sum_w (-1)^{\sum_{i\in U} w_i} (f^*(w))^2$.

$$\begin{aligned}
bs_d(f) &= \mathbf{E}[BS_{d,f}(w)] = \frac{1}{2^n}\sum_w BS_{d,f}(w) \\
&= \frac{1}{2^n}\sum_w \left[\sum_{i=1}^k \left(f(w) - f(w^{(B_i)})\right)^2\right] \\
&= \frac{1}{2^n}\sum_w \left[\sum_{i=1}^k \left(f(w) + f(w^{(B_i)}) - 2f(w)f(w^{(B_i)})\right)\right] \\
&= \sum_{i=1}^k \left[\frac{1}{2^n}\sum_w f(w) + \frac{1}{2^n}\sum_w f(w^{(B_i)}) - \frac{2}{2^n}\sum_w f(w)f(w^{(B_i)})\right] \\
&= 2\sum_{i=1}^k \left[\sum_w (f^*(w))^2 - \sum_w (f^*(w))^2 (-1)^{\sum_{j\in B_i} w_j}\right] \\
&= \sum_w (f^*(w))^2 \left[2\sum_{i=1}^k \left(1 - (-1)^{\sum_{j\in B_i} w_j}\right)\right],
\end{aligned}$$

where $(-1)^{\sum_{j\in B_i} w_j}$ measures the parity of w in B_i. Then, it follows

$$bs_d(f) = \sum_w p(w,d)\,(f^*(w))^2 ,$$

where the *weight function* $p(w,d)$ is given by:

$$p(w,d) = 2\sum_{i=1}^k \left(1 - (-1)^{\sum_{j\in B_i} w_j}\right) .$$

Analysis of $\boldsymbol{p(w,d)}$

Given a string $w \equiv (w_1,\ldots,w_n) \in \{0,1\}^n$, of cardinality $|w|$, we consider two different cases.

$\boldsymbol{|w| < k}$

We have $p(w,d) = 2\left[k - \sum_{i=1}^k (-1)^{\sum_{j\in B_i} w_j}\right]$ and $\sum_{i=1}^k (-1)^{\sum_{j\in B_i} w_j} \geq k - 2|w|$, where the minimum is attained when each of the $|w|$ bits equal to 1 in the string w belongs to a different set B_i. Thus, we obtain

$$p(w,d) = 2k - 2\sum_{i=1}^k (-1)^{\sum_{j\in B_i} w_j} \leq 4|w| .$$

$|\boldsymbol{w}| \geq \boldsymbol{k}$

In this case, we have $\sum_{i=1}^{k}(-1)^{\sum_{j\in B_i} w_j} \geq -k$ and this minimum is reached when the $|w|$ bits equal to 1 of the string w are distributed among the sets B_i in a way such that each B_i contains an odd number of ones. Thus, we obtain

$$p(w,d) = 2k - 2\sum_{i=1}^{k}(-1)^{\sum_{j\in B_i} w_j} \leq 2k + 2k = 4k\,.$$

By putting together these results, we have

$$p(w,d) \leq \begin{cases} 4k & k \leq |w|\,, \\ 4|w| & |w| < k\,, \end{cases}$$

i.e., $p(w,d) \leq 4\min\{|w|,k\} \leq 4|w|$ and $\max_{d\in\mathcal{D}} p(w,d) = 4|w|$. Moreover, this maximum is reached when one takes the decomposition of the set $\{1,2,\ldots,n\}$ in exactly n subsets of only one bit, i.e. $B_i = \{i\}$, $1 \leq i \leq n$. Finally, from the fact that $s(f) = 4\sum_w |w|\,(f^*(w))^2$ (see [12]), it follows that $bs_d(f) \leq s(f)$ and

$$\begin{aligned} \widehat{\beta s}_d(f) &= \max_{d\in\mathcal{D}} bs_d(f) \\ &= \max_{d\in\mathcal{D}} \sum_w p(w,d)\,(f^*(w))^2 \\ &= \sum_w (f^*(w))^2 \max_{d\in\mathcal{D}} p(w,d) \\ &= 4\sum_w |w|\,(f^*(w))^2 = s(f)\,. \end{aligned}$$

□

Since sensitivity and block sensitivity are two different complexity measures, the fact that by averaging one can cancel the difference between them is an indication that this is not the 'right' way of taking averages. In other words, this result shows how our previous definition of average block sensitivity, $bs(f)$, is more appropriate then the one considered here.

Chapter 4

Combinatorial Properties of certain Boolean functions

In this chapter we present some results concerning the connections between combinatorial properties of Boolean functions and their complexity.

We investigate the representations of Boolean functions as polynomials over R and over the field $GF(2)$, and prove an interesting characterization theorem: the degree of a Boolean function over $GF(2)$ is a measure of the size of its largest subfunction that takes the value 1 on an odd number of input strings (Section 4.1). We present some properties of *odd* functions, i.e. functions that take the value 1 on an odd number of strings (Section 4.2) and analyze the connections between the problem of the existence of odd functions with very low maximal sensitivity and the long standing open problem of the relationship between the maximal sensitivity and the block sensitivity of Boolean functions. We then prove a theorem regarding the degree of Boolean functions computable by read-once formulas (Section 4.3).

The second part of the chapter is devoted to the presentation of our main result: we show how the Fourier based techniques can be applied to derive new lower bounds on the size complexity of previously unclassified Boolean functions (Section 4.4). These bounds are related to AC^0 functions, and generalize the well known result stating that constant depth circuits require exponential size to compute the parity function.

One of the best-known results in Circuit Complexity is that constant depth circuits require exponential size to compute the parity function. Here we generalize this result to a class of functions, which we call *strongly balanced functions*.

We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *balanced* if it takes the value 1 on exactly half the input strings; *balanced up to the level k* (in short *k-balanced*) if any subfunction depending on k variables is balanced; and *strongly balanced* if f is balanced up to a level k such that $n - k = \omega\left((\log n)^d\right)$ for a constant $d > 0$. These definitions are made precise below. Our main result is that AC^0 circuits cannot compute strongly balanced functions. More precisely, we prove that a circuit of constant depth d require size $\Omega\left(2^{\frac{(n-k)^{1/d}}{20}}\right)$ to compute a k-balanced function. Note that this class of functions contains the parity and its complement as very special cases, since they are the only functions balanced up to $k = 1$. Moreover, it should be pointed out that this result allows us to gain insights to the structural properties of AC^0 functions.

The main tool of the proof is the harmonic analysis on the hypercube, that yields an interesting spectral characterization of k-balanced functions, together with the result proved in [45], stating that AC^0 functions have almost all of their *power spectrum* on the low-order coefficients.

Some of the results presented in this chapter also appeared in [11].

4.1 The Polynomial Representation

Over the last decade, several authors have employed polynomial methods in Circuit Complexity [8]. One of the goals of this approach is that of showing that Boolean functions computable by certain kinds of circuits can be represented by polynomials of a given type, e.g. low degree polynomials. These results have naturally provided lower bound techniques. For this reason, it could be interesting to study the properties of the representations of Boolean functions as polynomials over various fields. In this section we will investigate two kinds of representation of Boolean functions by polynomials: the representation over R and that over the field $GF(2)$, and we will prove an interesting characterization theorem about the degree of a Boolean function as a polynomial over $GF(2)$.

Definition 4.1 *Let f be a Boolean function on n variables and $p : F^n \to F$ a multivariate polynomial over a field F. We say that p* represents *f if, for every $w \in \{0,1\}^n$, $f(w) = p(w)$.*

Every Boolean function can be represented as a polynomial. Moreover, since $x^2 = x$ for all variables $x \in \{0,1\}$, the polynomials are automatically multilinear.

The basic parameter of the representation of a Boolean function as a polynomial is its *degree*.

Definition 4.2 *For a Boolean function f, the* degree *of f over a field F, denoted by $d_F(f)$, is the degree of the unique multilinear polynomial over F that represents f.*

In the following analysis, we will consider the degree of a function f as a polynomial over R and as a polynomial over $GF(2)$.

Definition 4.3

- *For a Boolean function f, the degree of f over R, denoted by $d_R(f)$, is the degree of the unique multilinear real polynomial that represents f exactly.*

- *For a Boolean function f, the degree of f over $GF(2)$, denoted by $d_2(f)$, is the degree of the unique multilinear polynomial over $GF(2)$ that represents f exactly.*

Note that the degree of a function f over the reals coincides exactly with the notion of degree defined as the maximum order of the nonzero Fourier coefficients (see Section 2.1).

The degree of a function over the reals can be related to several complexity measures of the function, such as the *decision tree complexity*, the *maximal sensitivity*, the *block sensitivity* and the *certificate complexity* (see Section 3.1.1 for their definitions). For istance, for any Boolean function f, we have $d_R(f) \leq D(f)$ and $d_R(f) \geq \sqrt{BS(f)}$ (see [67]).

With respect to the degree of a function over the fields $GF(m)$, some interesting lower bounds can be found in [69].

4.1.1 Polynomials over R and $GF(2)$

Let $x_i \in \{0,1\}$, $i = 1, 2, \ldots, n$. For each set $B \subseteq \{1, 2, \ldots, n\}$, we consider the term

$$\prod_{i \in B} x_i \prod_{j \notin B} (1 - x_j) .$$

Note that the term associated to the set $\{1, 2, \ldots, n\}$ is given by $\prod_{i=1}^{n} x_i$, while the term associated to the empty set is $\prod_{j=1}^{n}(1 - x_j)$. There is a one-to-one correspondence between terms (associated to sets B) and Boolean strings $w \in \{0,1\}^n$: $w_i = 1 \iff i \in B$. These 2^n terms are a basis for the 2^n-dimensional vector space of functions from $\{0,1\}^n$ into the field R of real numbers. Therefore, any function $f : \{0,1\}^n \to R$ (and thus any Boolean function) can be written as a linear combination of these terms. The coefficients of the linear combination are just the values of f. Precisely, the coefficient of a given term is equal to the value that the function takes on the string corresponding to that term:

$$f(x_1, \ldots, x_n) = \sum_{B \subseteq \{1, \ldots, n\}} f(B) \prod_{i \in B} x_i \prod_{j \notin B} (1 - x_j) .$$

This representation of the function f is called *table look-up representation* (see [8]). The *standard polynomial representation* of f is then obtained by multiplying out the table look-up representation via the distributive law. Consequently, the monomials of the standard polynomial representation are represented by the 2^n terms

$$\prod_{i \in B} x_i \,,$$

where $B \subseteq \{1,,\ldots,n\}$, which are a basis for the vector space of functions from $\{0,1\}^n$ into R, too. Notice that, even in this case, there is a one-to-one correspondence between terms and Boolean strings.

There exists a very convenient matrix formulation that allows us to compute (in at most exponential time) the vector $\hat{f}$ of the coefficients of the standard representation.

Let f be a Boolean function depending on n variables, and let L_n be a $2^n \times 2^n$ matrix recursively defined as follows

$$L_1 = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, \qquad L_n = \begin{pmatrix} L_{n-1} & 0 \\ -L_{n-1} & L_{n-1} \end{pmatrix}.$$

Lemma 4.1 *The coefficients of the standard polynomial representation of f are the entries of the vector $\hat{f}$, which is given by*

$$\hat{f} = L_n f \,.$$

Proof By induction on the number of variables.

Base

Let $n = 1$. The standard representation is given by

$$p = f(0)(1 - x_1) + f(1)x_1 = f(0) + [f(1) - f(0)]x_1 \,,$$

and, by direct inspection, one can check that the two coefficients are given by

$$\hat{f} = L_1 f = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}.$$

Induction Step

Let $f : \{0,1\}^n \to \{0,1\}$. This function can be viewed as the concatenation of two functions, each depending on $n - 1$ variables

$$f = \begin{pmatrix} f_0 \\ f_1 \end{pmatrix},$$

where

$$\begin{aligned} f_0(x_2, x_3, \ldots, x_n) &= f(0, x_2, x_3, \ldots, x_n) \\ f_1(x_2, x_3, \ldots, x_n) &= f(1, x_2, x_3, \ldots, x_n) \,. \end{aligned}$$

By the induction hypothesis, we know that the coefficients of the two polynomial representations of f_0 and f_1, p_0 and p_1, are given by

$$\begin{aligned}\hat{f}_0 &= L_{n-1} f_0\\ \hat{f}_1 &= L_{n-1} f_1\,.\end{aligned}$$

To conclude, observe that the polynomial p representing f can be computed from p_0 and p_1 as follows:

$$p = (1 - x_1)p_0 + x_1 p_1\,.$$

Therefore, we get

$$\begin{aligned}\hat{f} &= \begin{pmatrix} I & 0 \\ -I & I \end{pmatrix}\begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \end{pmatrix} = \\ &= \begin{pmatrix} I & 0 \\ -I & I \end{pmatrix}\begin{pmatrix} L_{n-1} & 0 \\ 0 & L_{n-1} \end{pmatrix}\begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \\ &= \begin{pmatrix} L_{n-1} & 0 \\ -L_{n-1} & L_{n-1} \end{pmatrix}\begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = L_n f\,.\end{aligned}$$

□

Observe that any row of the matrix L_n is related to a monomial. Precisely, the i-th row is related to the monomial $\prod_{i\in B} x_i$ associated to the set B of the integers j such that the j-th bit in the binary expansion of i is equal to 1. Moreover, from the definition of the matrix L_n, it follows that every row has exactly one-half of its nonzero entries equal to 1 and the other half equal to -1. Finally, the last row of L_n, which is related to the monomial $x_1 x_2 \dots x_n$, coincides with the truth-table of the parity function of n variables, where the values $\{1, -1\}$ are used instead of $\{0, 1\}$.

Let us now turn to the polynomial representation of Boolean functions over $GF(2)$.

Definition 4.4 ([67]) *The polynomial representation of Boolean functions over the field $GF(2)$ is given by the polynomial in the real case, except the coefficients are taken modulo 2.*

From this definition, it follows that the polynomial representation of a function over $GF(2)$ can be obtained by simply keeping those monomials with an odd coefficients from its real polynomial representation, and by giving them the value 1 as new coefficient. Thus, the vector $\hat{f}_2$ of the coefficients of the polynomial over $GF(2)$, representing a Boolean function $f : \{0,1\}^n \to \{0,1\}$, can be computed as follows:

$$\hat{f}_2 = L_n\, f \pmod 2\,.$$

Moreover, since $-1 \equiv 1 \pmod 2$, we can recursively define a new matrix M_n,

$$M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad M_n = \begin{pmatrix} M_{n-1} & 0 \\ M_{n-1} & M_{n-1} \end{pmatrix},$$

such that

$$\hat{f}_2 = M_n\, f \pmod 2 .$$

Finally, note that $d_2(f) \leq d_R(f)$.

4.1.2 Odd and Even Boolean functions

We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *odd* if it takes the value 1 on an odd number of strings $w \in \{0,1\}^n$. More precisely, we have the following definition.

Definition 4.5 *A function f is* odd *if and only if the cardinality of the set $\{w \in \{0,1\}^n \mid f(w) = 1\}$ is expressed by an odd number. Conversely, we say that a function f is* even *if the cardinality of the set $\{w \in \{0,1\}^n \mid f(w) = 1\}$ is expressed by an even number.*

We will use the notation $|f|$ to denote the number of strings accepted by the function f, i.e. $|f| = |\{w \in \{0,1\}^n \mid f(w) = 1\}|$.

The fact that a function is odd has some 'weird' implications. For instance any odd function has the maximum possible degree over the reals:

Theorem 4.1 *For any odd function $f : \{0,1\}^n \to \{0,1\}$, $d_R(f) = n$.*

Proof From the structure of the matrix L_n it follows that the coefficient of the maximum degree monomial $x_1 x_2 \ldots x_n$ can take value zero if and only if exactly one half of the strings w accepted by f are such that $parity(w) = -1$ and the other half are such that $parity(w) = 1$. But this cannot happen if f takes the value 1 on an odd number of strings.

□

While in Section 4.2 we will analyze other properties of odd functions, here we focus our attention on the class of even functions and enforce their definition introducing the notion of functions *even up to the level k*. In order to do this, we will make use of the notion of *restriction* (see Section 1.4). Let $0 \leq k \leq n$.

Definition 4.6 *We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is* even up to k *if the following two conditions hold:*

- *For any restriction ρ such that $size(\rho) = k$, the resulting subfunction f_ρ is an* even *function.*

- *There exists a restriction* μ *with* $size(\mu) = k - 1$ *such that* f_μ *is an* odd *function.*

We now prove a technical lemma which will be useful later in this section.

Lemma 4.2 *Let* f *be a Boolean function even up to* k*, and let* μ *denote the restriction, of size* $k-1$*, defining its largest* odd *subfunction. Then, all the restrictions* δ *of size* $k-1$*, such that* $dom(\mu) = dom(\delta)$ *define odd subfunctions.*

Proof Let the restriction μ, of size $k-1$, define an odd subfunction. Pick any variable, say x_j, from the set $dom(\mu)$ and define another restriction $\hat{\mu}$ as follows

$$\hat{\mu}(x_i) = \begin{cases} \mu(x_i) & \textit{if } i \neq j \\ 1 - \mu(x_i) & \textit{if } i = j\,. \end{cases}$$

Since by concatenating f_μ and $f_{\hat{\mu}}$ we get a new subfunction depending on k variables, from the fact that f is even up to k, it follows that $f_{\hat{\mu}}$ must also be odd. Then, the thesis follows by repeating the same argument until one has considered all the possible assignments to the variables in the set $dom(\mu)$.

□

4.1.3 The Characterization Theorem

We now prove a theorem that connects the degree of a Boolean function over $GF(2)$ with the concept of *evenness* up to a certain level. More precisely, we will prove that the degree over $GF(2)$ is a measure of the size of the largest odd subfunction of a Boolean function f.

We first prove two technical lemmas, one about the structure of the matrix M_n and the other one about the meaning of the coefficients over $GF(2)$. In the following, we will index the rows and the columns of M_n by using binary strings $w, \mu \in \{0,1\}^n$, taken in the usual lexicographic order, instead of integers $i, j = 0, 1, \ldots, 2^n - 1$. Moreover, we will denote by $\hat{f}_2(w)$ the coefficient of the monomial $\prod_{i \in W} x_i$, where $W \equiv \{i \mid w_i = 1\}$.

Lemma 4.3 *For every* $w, \mu \in \{0,1\}^n$*, we have*

$$M_n(w,\mu) = \begin{cases} 1 & \text{if } \quad \forall\, i \;\; (\; w_i = 0 \;\Rightarrow\; \mu_i = 0 \;) \\ 0 & \text{otherwise}\,. \end{cases} \qquad (1)$$

Proof By induction on n.

Base

For $n = 1$, we have

$$M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

that is $M_1(0,0) = 1$, $M_1(0,1) = 0$, $M_1(1,0) = 1$ and $M_1(1,1) = 1$, according to our thesis.

Induction step

We know that M_n can be recursively defined as follows:

$$M_n = \begin{pmatrix} M_{n-1} & 0 \\ M_{n-1} & M_{n-1} \end{pmatrix}.$$

Let $w \equiv (w_1, w_2, \ldots, w_n) \in \{0,1\}^n$ and $\mu \equiv (\mu_1, \mu_2, \ldots, \mu_n) \in \{0,1\}^n$. We consider the four submatrices separately.

- The rows and the columns in the first submatrix are indexed by strings such that $w_1 = \mu_1 = 0$. Then, these strings satisfy condition (1), and the lemma follows by induction hypothesis.
- The second submatrix has all zero entries, and this agrees with the lemma because its rows and columns are indexed by strings such that $w_1 = 0$ and $\mu_1 = 1$.
- For the third and the fourth submatrices, the lemma follows by induction hypothesis since condition (1) is obviously satisfied ($w_1 = 1$).

□

Note that for every $w \in \{0,1\}^n$, the related row in M_n, which we denote by $M_n(w)$, has exactly $2^{|w|}$ entries equal to 1.

Now, let us associate a restriction ρ_w, of size $|w|$, to any row $M_n(w)$ in the following way:

- $\rho_w(x_i) = 0$ for any i such that $w_i = 0$;
- $\rho_w(x_i) = \star$ for any i such that $w_i = 1$.

This allows us to relate each row of M_n to a particular subfunction of f, i.e. f_{ρ_w}.

Lemma 4.4 *For every* $w \in \{0,1\}^n$, $\hat{f}_2 = |f_{\rho_w}| \pmod 2$.

Proof The coefficient $\hat{f}_2(w)$ is given by the scalar product between the row $M_n(w)$ and the truth-table of the function f (mod 2). The lemma follows from the fact that the $2^{|w|}$ nonzero entries of $M_n(w)$ are exactly located in the positions corresponding to the values defining the truth-table of the subfunction f_{ρ_w}.

□

We are now able to state and prove the following *Characterization Theorem*.

Theorem 4.2 *Let f be a Boolean function depending on n variables. Then, the following two statements are equivalent:*

1. $d_2(f) = k$;
2. *f is even up to level $k+1$.*

Proof We first prove that if a function f has degree k over $GF(2)$, then it is even up to the level $k+1$. Since $d_2(f) = k$, Lemma 4.4 implies that f has an odd subfunction of size k. Thus, to prove that f is even up to $k+1$, we only have to prove that for any restriction δ of size $k+1$, f_δ is even.

Let w be the string of cardinality $|w| = k$, such that f_{ρ_w} is odd. Observe that, for any μ of size $k+1$ defined by taking the all zero assignment for the variables in its domain, the subfunction f_μ is even, as follows from Lemma 4.4 and from the fact that $d_2(f) = k$.

For any such restriction, we define another restriction δ, by picking a variable from $dom(\mu)$ and by changing its value from 0 to 1. By concatenating the two subfunctions f_μ and f_δ we then get a new subfunction depending on $k+2$ variables, which can be obtained from f by assigning the value 0 to $n-(k+2)$ variables. Since this subfunction is related to a monomial of degree $k+2$ in the standard polynomial representation of f, it must be even. Thus, it follows that f_δ must also be even.

The thesis then follows by repeating the same argument until one has considered all the possible assignments to the variables in $dom(\mu)$.

Conversely, from the definition of functions even up to the level k, and from Lemma 4.2, it follows that, for any restriction ρ of size $k+1$, the subfunction f_ρ is even, while there must exist a string v, of cardinality $|v| = k$, such that f_{ρ_v} is odd. Then, the theorem immediately follows from Lemma 4.4.

□

This theorem allows us to derive a clear and interesting combinatorial characterization of the degree over $GF(2)$.

Corollary 4.1 *The degree of the polynomial representation of a Boolean function over the field $GF(2)$ is equal to the size of its largest odd subfunction.*

□

Moreover, the Characterization theorem has another interesting consequence: only odd functions can have maximum degree over $GF(2)$.

Corollary 4.2 *A Boolean function f on n variables is odd if and only if $d_2(f) = n$.*

□

Finally, we obtain a result related to the degree of a function over R:

Corollary 4.3 *If a function f is even up to the level k, then $d_R(f) \geq k$.*

Proof Follows from $d_R(f) \geq d_2(f)$.

□

4.2 Properties of Odd functions

In this section we present some properties of odd functions, and analyze the connection between the problem of the existence of odd functions with very low maximal sensitivity and the long standing open problem of the relationship between maximal and block sensitivity of Boolean functions.

Remember that we say that a function is odd if it takes the value 1 on exactly half of the input strings. There exist 2^{2^n-1} odd functions on n variables, that is exactly half of all Boolean functions on n variables are odd.

The fact that a function is odd has several implications. Let $w^{(i)}$ denote the string obtained from a string w by flipping the i-th bit.

Definition 4.7 f depends *on dimension i if there exists an input vector w such that $f(w) \neq f(w^{(i)})$. f is called* nondegenerated *if it depends on all n dimensions.*

Proposition 4.1 *Odd functions are nondegenerated, that is they depend on all their variables.*

Proof We have to show that for every dimension i, $1 \leq i \leq n$, there exists $w \in \{0,1\}^n$ such that $f(w) \neq f(w^{(i)})$.

We prove this proposition by contradiction. Let us suppose that there exists a dimension i such that, for every $w \in \{0,1\}^n$, $f(w) = f(w^{(i)})$. This means that the two subfunctions obtained from f by assigning the values 0 and 1 to the i-th variable are equal, i.e. $f_{x_i \leftarrow 0} \equiv f_{x_i \leftarrow 1}$. Then $|f_{x_i \leftarrow 0}| = |f_{x_i \leftarrow 1}|$, and since

$$|f| = |f_{x_i \leftarrow 0}| + |f_{x_i \leftarrow 1}|,$$

it follows that f cannot be an odd function.

$\square$

Proposition 4.2 *Let f be an odd function on n variables. Then, for any $w \in \{0,1\}^n$, the Fourier coefficient $f^*(w)$ is nonzero.*

Proof Let us suppose that there exists a string w such that $f^*(w) = 0$. From the definition of $f^*(w)$ we have

$$f^*(w) = \frac{1}{2^n} \sum_x (-1)^{x^T w} f(x).$$

Moreover, from the fact that the Fourier kernel function $(-1)^{x^T w}$ takes the value 1 on exactly 2^{n-1} strings, and value -1 on the others 2^{n-1}, it follows that $f^*(w)$ can take value zero if and only if exactly one half of the strings x accepted by f are such that $(-1)^{x^T w} = 1$ and the other half are such that $(-1)^{x^T w} = -1$. But this implies that f must take the value 1 on an even number of strings.

$\square$

Notice that from this proposition it follows that odd functions have the maximum possible degree over the reals, as we already know since odd functions have maximum degree over $GF(2)$ (see Section 4.1.3).

Proposition 4.3 *For any odd function f, $d_R(f) = n$.*

We now consider other measures of complexity for Boolean functions and analyze the value that they take on the class of odd functions. We start with the decision tree complexity $D(f)$.

Proposition 4.4 *For any odd function f, $D(f) = n$.*

Proof Follows from the inequality $d_R(f) \leq D(f)$ and from Proposition 4.3. It is possible to give a direct proof of this result, that is a proof that uses in a direct way the hypothesis that f is odd. Let us suppose that $D(f) = k < n$. Then, in order to decide the value of the function, it is sufficient to read k bits out of n, in the worst case. This implies that the function will take the same value on the 2^{n-k} strings that agree on these k bits. In all the other cases, it is sufficient to read a number of bits less or equal to k, and from the same argument as before, we can conclude that the function always take the same value on 2^{n-h} strings, where $h \leq k$. But from this argument it follows that the function must be even.

□

Let us now consider the certificate complexity $C(f)$.

Proposition 4.5 *For any odd function f, $C(f) \geq \sqrt{n}$.*

Proof Follows from $D(f) \leq C^2(f)$ (see Lemma 3.3) and from Proposition 4.4.

□

The lower bound on the certificate complexity of odd functions is tight and it is achieved, e.g., by the following odd function

$$h = \bigwedge_{i=1}^{\sqrt{n}} g_i \quad \text{with} \quad g_i = \bigvee_{j=1}^{\sqrt{n}} x_{i_j} \, .$$

We now turn to the block sensitivity.

Proposition 4.6 *For any odd function f, $BS(f) \geq \sqrt[4]{n}$.*

Proof Follows from $C(f) \leq BS^2(f)$ (see Lemma 3.2) and from Proposition 4.5.

□

Since for monotone functions we have $S_{max}(f) = BS(f) = C(f)$ (see Proposition 3.1), from Proposition 4.5 it follows that odd monotone functions must have high maximal sensitivity, that is $S_{max}(f) \geq \sqrt{n}$. Note that this bound is tight and it is again reached by the odd monotone function h defined before. On the contrary, there exist even monotone functions with very low maximal sensitivity, e.g., the *monotone address function*, which has maximal sensitivity $O(\log n)$ (see [73]).

From a combinatorial point of view, it would be very interesting to understand how the oddness of a monotone function could raise the maximal sensitivity up to $\sqrt{n}$. More in general, it would be interesting to find a direct proof of the lower bound $BS(f) \geq \sqrt[4]{n}$, for f odd, and to verify whether this bound is tight.

4.2.1 Maximal Sensitivity of Odd functions

We now analyze the connections between the problem of the existence of odd functions with very low maximal sensitivity and the long-standing open problem of the relationship between maximal and block sensitivity of Boolean functions (see Chapter 3). We also introduce the notions of *totally-even*, *strongly-even* and *quasi-odd* functions and characterize their block sensitivity.

First of all, we state the following easy lemma, which shows how the existence of an odd function with polylogarithmic maximal sensitivity would imply the existence of an exponential gap between maximal and block sensitivity.

Lemma 4.5 *Let $f : \{0,1\}^n \to \{0,1\}$ be an odd Boolean function. If*

$$S_{max}(f) = (\log n)^{O(1)},$$

then f presents an exponential gap between its maximal sensitivity and its block sensitivity.

Proof The thesis easily follows from the fact that for any odd function f, $BS(f) \geq \sqrt[4]{n}$.

$\square$

Note that the existence of an odd function with a superpolynomial gap between maximal and block sensitivity implies the existence of an even function with the same property. In fact, if $f : \{0,1\}^n \to \{0,1\}$ has this property, then by concatenating f and its complement $\neg f$, we get a nondegenerated even function $h : \{0,1\}^{n+1} \to \{0,1\}$, such that $S_{max}(h) = S_{max}(f) + 1$ and $BS(h) = BS(f) + 1$.

Lemma 4.5 suggests how the problem of finding a nontrivial lower bound on the maximal sensitivity of odd functions should be quite difficult. In fact, by exibiting an odd function with polylogarithmic maximal sensitivity, we will immediately solve the long-standing open problem of the relationship between maximal and block sensitivity, which is considered a difficult one. Anyway, it would be interesting to understand in a deeper way the connections between these two problems. In particular, we would like to understand if these two problems are equivalent, that is if the existence of an even function with a superpolynomial gap between maximal and block sensitivity would imply the existence of an odd function with very low maximal sensitivity. To this aim, let us define the following subclasses of even Boolean functions:

Definition 4.8 *Let f be a Boolean function depending on n variables. We say that:*

- *f is* totally-even *if $d_2(f)$ is a constant;*
- *f is* strongly-even *if $d_2(f) = o(n)$;*
- *f is* quasi-odd *if $d_2(f) \geq n - n^{\varepsilon}$, where $0 < \varepsilon < 1$.*

The problem of the existence of a superpolynomial gap between S_{max} and BS is equivalent to the problem of the existence of odd functions with very low maximal sensitivity under the following conjecture:

Conjecture *An even function f can present a superpolynomial gap between its maximal sensitivity and its block sensitivity if and only if f is of the* quasi-odd *type.*

We prove the equivalence of the two problems, under this conjecture, in the following lemma.

Lemma 4.6 *If there exists a quasi-odd function f such that there is no polynomial relationship between its maximal and block sensitivity, then there exist odd functions with very low maximal sensitivity.*

Proof Easily follows by considering the odd subfunction f_μ of size of order $n - n^\varepsilon$. In fact, $S_{max}(f_\mu) \leq S_{max}(f)$ and $BS(f_\mu) \geq \sqrt[4]{n - n^\varepsilon}$ (see Proposition 4.6).

□

Finally, we prove that totally-even Boolean functions cannot present a superpolynomial gap between maximal and block sensitivity. We will use a result proved in [40].

Definition 4.9 *Given an integer ℓ, the ℓ-*block sensitivity *of a function f on a string w is defined to be the largest number t such that there exists t* disjoint *sets $B_1, B_2, \ldots, B_t$ such that, for any $1 \leq i \leq t$,*

- $|B_i| \leq \ell$;
- *f is sensitive to B_i on w.*

*The ℓ-*block sensitivity *of f, $BS_\ell(f)$, is the maximum, over all w, of the ℓ-block sensitivity of f on w.*

Theorem 4.3 ([40]) *For any Boolean function f, $BS_\ell(f) = O\left(S_{max}(f)^\ell\right)$.*

□

Lemma 4.7 *For any totally-even function f, $BS(f) = (S_{max}(f))^{O(1)}$.*

Proof Observe that for any totally-even function f we have $BS(f) = BS_\ell(f)$, where ℓ is the constant which exactly defines the degree of f over $GF(2)$. In other words, this means that a totally-even function can be sensitive only to blocks of constant (minimal) size.

To see why this fact holds, let us consider any minimal-size block B such that f is sensitive to B on a string $w = (w_1, w_2, \ldots, w_n)$. The following two conditions hold

- $f(w) \neq f(w^{(B)})$;
- for any $C \subset B$, $f(w) = f(w^{(C)})$.

Now, observe that these conditions imply that the subfunction f_β, where β is the following restriction

$$\beta(x_i) = \begin{cases} w_i & \text{if } i \notin B \\ \star & \text{if } i \in B\,, \end{cases}$$

is an odd function. More precisely, it turns out that $|f_\beta| = 1$.

Finally, from the fact that $d_2(f) = \ell$, it follows that $|B| \leq \ell$.

□

Lemma 4.7 could be considered as a weak support to the conjecture that among the class of even functions, only the functions of the quasi-odd type could present a superpolynomial gap between maximal and block sensitivity. Clearly, to settle this conjecture, we should be able to understand what happens to the block sensitivity of strongly-even functions. Unfortunately, at the moment this question seems to be as difficult as the 'sensitivity versus block sensitivity' problem

4.3 Read Once Boolean functions

We now prove a theorem regarding the degree of Boolean functions computable by read-once formulas (read-once functions from now). Recall that a *formula* is a Boolean circuit of fan-out 1 and a *read-once formula* is a formula in which each variable appears only once. These functions are important, especially for "low" complexity classes, because every NC^1 function on n variables can be viewed as the projection of a read-once function with $n^{O(1)}$ variables (see [63] and [70] for the notion of projection). The theorem we prove leads to the following combinatorial property: nondegenerated read once functions are odd. That is, they can accept only an odd number of strings.

Theorem 4.4 *For any nondegenerated read once Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ we have*

$$d_R(f) = d_2(f) = n\,.$$

Proof We will prove that $d_2(f) = n$ by induction on the number n of variables. The result for $d_R(f)$ easily follows since, for any Boolean function f, $d_2(f) \leq d_R(f)$.

Base

For $n = 1$ the function f is simply $f = x_1$ (or $f = \neg x_1$), and its polynomial representation over $GF(2)$ is given by $p(x_1) = x_1$ (or $p(x_1) = 1 - x_1$), respectively. Then, the thesis follows since in both cases we get $d_2(f) = 1$.

Induction Step

Let $f : \{0,1\}^n \rightarrow \{0,1\}$. Without loss of generality we can assume that f is obtained starting from two functions f_1 and f_2 depending on n_1 and n_2

variables, respectively. In order to get a read once function f, f_1 and f_2 must also be read once, and they must depend on two disjoint subsets of variables. Therefore we have

$$n = n_1 + n_2 \,.$$

Moreover, by induction hypothesis, we know that $d_2(f_1) = n_1$ and $d_2(f_2) = n_2$. Let p_1 and p_2 denote the polynomial representations over $GF(2)$ of f_1 and f_2, respectively.

We now consider two different cases.

$\boldsymbol{f \equiv f_1 \wedge f_2}$

The polynomial representation p of f is simply given by

$$p = p_1 \cdot p_2 \,.$$

Note that, since we are considering polynomials over $GF(2)$, by the induction hypothesis we know that the coefficients of the two maximum degree monomials in p_1 and p_2 must be equal to 1. This implies that the coefficient of the monomial given by the product of these two monomials will also be equal to 1. Thus we get

$$d_2(f) = d_2(f_1) + d_2(f_2) = n_1 + n_2 = n \,.$$

$\boldsymbol{f \equiv f_1 \vee f_2}$

In this case, the polynomial representation p of f is given by

$$p = p_1 + p_2 - p_1 \cdot p_2 \,,$$

and, by the same argument as before, we can conclude that

$$d_2(f) = d_2(f_1) + d_2(f_2) = n_1 + n_2 = n \,.$$

□

Corollary 4.4 *Nondegenerated read once Boolean functions are odd.*

Proof The degree of a Boolean function over $GF(2)$ can be characterized as the size of the largest odd subfunction of f. Therefore, $d_2(f) = n$ if and only if f is an odd function. The corollary, then follows immediately from Theorem 4.4.

□

Observe that read once functions can be regarded as monotone functions. Indeed, without loss of generality, we can assume that the circuit computing a read once function f has alternating levels of AND and OR gates, and no negation gates: all negation gates can be pushed to be applied to input only, and renaming all negative literals as positive ones does not change properties of read once functions.

Therefore, from Proposition 3.1 and from Proposition 4.5 it follows that for any read once function f depending on n variables

$$S_{max}(f) = BS(f) = C(f) \geq \sqrt{n}\,.$$

This lower bound is tight and it is reached, e.g., by the function

$$h = \bigwedge_{i=1}^{\sqrt{n}} g_i \quad \text{with} \quad g_i = \bigvee_{j=1}^{\sqrt{n}} x_{i_j}\,.$$

4.4 The Class of Balanced Boolean functions

One of the best-known results in Circuit Complexity is that constant depth circuits require exponential size to compute the parity function (see [33] and [35]). Here we generalize this result to a class of functions, which we call *strongly balanced functions*. Our main tool is the harmonic analysis on the hypercube, that yields an interesting spectral characterization of the class of functions under investigation, together with the result proved in [45], stating that AC^0 functions have almost all of their *power spectrum* on the low-order coefficients (see Section 2.7 for more details).

Let f be a Boolean function depending on n variables. We say that f is *balanced* if the probability that f takes the value 1 is equal to $\frac{1}{2}$.

Definition 4.10 *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is* balanced *if $|f| = 2^{n-1}$, that is if it takes the value 1 on exactly half of the input strings.*

We would like to enforce this definition introducing the notion of functions *balanced up to the level k* (shortly *k-balanced* functions). In order to do this, we will make use of the notion of *restriction* (as we have already done for the class of even functions). Let $1 \leq k \leq n$.

Definition 4.11 *We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is* balanced up to the level k*, if for any restriction ρ such that $size(\rho) = k$, the resulting subfunction f_ρ is balanced.*

Observe that if a function is k-balanced, then it is also ℓ-balanced, for any $\ell \geq k$, since the fact that f is k-balanced does not imply that its restrictions of size $k-1$ are not balanced (notice the difference with the definition of function even up to k).

We finally define the class of *strongly balanced* Boolean functions.

Definition 4.12 *A Boolean function f depending on n variables is* strongly balanced *if there exists a constant d such that f is balanced up to a level k s.t. $n - k = \omega\left((\log n)^d\right)$.*

It is very important to note that, for any k between 1 and n, there exist nondegenerated functions which are k-balanced, but are not $(k-1)$-balanced. We will provide some examples later.

4.4.1 Spectral Characterization and Size-complexity

Our main result states that strongly balanced functions cannot be computed by AC^0 circuits. In order to prove it, we first derive a spectral characterization of k-balanced functions, and then use it, together with Lemma 2.11, to determine a lower bound on the size required by a depth d circuit to compute them. Finally, an easy application of this bound will provide our thesis.

Theorem 4.5 *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is balanced up to the level k if and only if the following two properties hold:*

(1) f is balanced;

(2) for any string w such that $0 < |w| \le n-k$, $f^*(w) = 0$.

Proof

- If f is k-balanced, then property (1) holds by definition. Thus, we only need to prove property (2).

  Let $\mu \equiv (\mu_1, \mu_2, \ldots, \mu_n)$ be a Boolean string such that $0 < |\mu| = n-\ell \le n-k$. Moreover, let $U = \{i \mid \mu_i = 1\}$. For any string $u \in \{0,1\}^{n-\ell}$, let f_u denote the subfunction defined by the restriction that assigns to the variables x_i such that $i \in U$, the $(n-\ell)$ values taken from the string u, and leaves undetermined the other ℓ variables.

  Then, we have

$$\begin{aligned} f^*(\mu) &= \frac{1}{2^n}\sum_w (-1)^{\mu^T w} f(w) \\ &= \frac{1}{2^n}\sum_w (-1)^{\sum_{i\in U} w_i} f(w) \\ &= \frac{1}{2^n}\sum_{u\in\{0,1\}^{n-\ell}} \left[(-1)^{|u|} \sum_{v\in\{0,1\}^{\ell}} f_u(v)\right] \\ &= \frac{1}{2^n}\sum_{u\in\{0,1\}^{n-\ell}} \left[(-1)^{|u|}\ |f_u|\right]. \end{aligned}$$

  For any $u \in \{0,1\}^{n-\ell}$, the subfunction f_u depends on $\ell \ge k$ variables and, since f is balanced up to k, we have $|f_u| = 2^{\ell-1}$. Thus, we get

$$f^*(\mu) = \frac{2^{\ell-1}}{2^n}\sum_{u\in\{0,1\}^{n-\ell}} (-1)^{|u|} = 0\,.$$

- We now prove that if properties (1) and (2) hold, then f is k-balanced.

  Let us choose $(n-k)$ variables out of n, and let U be the set of the indices of these $(n-k)$ variables. For any $u \in \{0,1\}^{n-k}$, let f_u denote the subfunction obtained from f by assigning to the variables in the set U, the $(n-k)$ values taken from the string u, and leaving undetermined the other k variables.

For any u, f_u depends on k variables. We show that any such subfunction accepts exactly 2^{k-1} inputs, i.e. for any string u, $|f_u| = 2^{k-1}$.

Let $f_{\#}$ denote the vector of the cardinality of the 2^{n-k} subfunctions f_u, and let f_U^* denote the vector of the Fourier coefficients related to the 2^{n-k} strings $w \equiv (w_1, w_2, \ldots, w_n)$ such that $w_i = 0$ for any $i \notin U$. Note that all the 2^{n-k} coefficients in the vector f_U^* are of order less or equal to $n-k$. Because of the recursive definition of Hadamard matrices, it turns out that

$$f_U^* = \frac{1}{2^n}\ H_{n-k}\ f_{\#}\,.$$

From properties (1) and (2), from the fact that the zero order Fourier coefficient is equal to the probability that the function takes the value 1, it then follows

$$f_U^* = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

from which

$$\begin{aligned} f_{\#} &= 2^n\ H_{n-k}^{-1}\ f_U^* = \frac{2^n}{2^{n-k}}\ H_{n-k}\ f_U^* \\ &= 2^{k-1}\ H_{n-k}\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 2^{k-1}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \end{aligned}$$

Thus, the theorem follows by repeating the same argument for all the $\binom{n}{k}$ choices of the set U.

□

This spectral characterization allows us to conclude that the parity function and its complement represent special cases among the class of k-balanced functions, since they are the only two functions balanced up to $k = 1$.

Corollary 4.5 *The only two functions balanced up to $k = 1$ are the parity function and its complement.*

Proof Let f be a function balanced up to the level 1, and let f_0^* denote the zero-order Fourier coefficient. Then from Parseval's identity and from Theorem 4.5, it follows that the Fourier coefficients of f must have the following absolute values:

$$|f^*(w)| = \begin{cases} \frac{1}{2} & \text{if} \quad |w| = 0 \\ 0 & \text{if} \quad 0 < |w| < n \\ \frac{1}{2} & \text{if} \quad |w| = n\,. \end{cases}$$

Thus, the corollary follows, since the only functions with such a kind of spectrum are the parity and its complement (see [45]).

□

At this point we are able to state and prove a lower bound on the size needed by a depth d circuit to compute a k balanced function, establishing in this way a connection between complexity and structural properties of Boolean functions.

Theorem 4.6 *Let f be a k-balanced Boolean function depending on n variables computable by a circuit of constant depth d and size M. Then*

$$M \geq \frac{1}{2} 2^{\frac{(n-k)^{1/d}}{20}}\,.$$

Proof An application of Lemma 2.11 yields the following inequality:

$$M \geq 2^{\frac{t^{1/d}}{20}}\, 2 \sum_{|w|>t} (f^*(w))^2\,.$$

Let us choose $t = n - k$. From the fact that $f^*(w) = 0$ for any $0 < |w| \leq n - k$, it follows

$$\sum_{|w|>n-k} (f^*(w))^2 = \sum_{w:|w|\neq 0} (f^*(w))^2 = \sum_{w} (f^*(w))^2 - (f_0^*)^2\,,$$

where f_0^* denotes the zero-order Fourier coefficient. Then, by using Parseval's identity $\sum_v (f^*(v))^2 = f_0^* = \frac{1}{2}$, we get $\sum_{|w|>n-k} (f^*(w))^2 = \frac{1}{4}$, and the thesis immediately follows:

$$M \geq 2^{\frac{(n-k)^{1/d}}{20}}\, 2 \sum_{|w|>t} (f^*(w))^2 = \frac{1}{2} 2^{\frac{(n-k)^{1/d}}{20}}\,.$$

□

Our main result, stating that strongly balanced functions do not belong to the set $AC^0[d]$, follows immediately as a corollary of Theorem 4.6.

Corollary 4.6 *A strongly balanced function f requires superpolynomial size to be computed by a depth d circuit.*

Proof Easily follows from Definition 4.12 and Theorem 4.6.

□

Note that the lower bound on the size can become exponential.

Corollary 4.7 *Depth d circuits require exponential size to compute k-balanced functions whenever k is s.t. $n - k = \Omega(n^{\varepsilon})$, for any positive constant $\varepsilon < 1$.*

Proof Immediate from Theorem 4.6.

□

We conclude this section by evaluating the average sensitivity of k-balanced functions.

Lemma 4.8

1. *If f is balanced up to $k < n$, then $s(f) > n - k$.*
2. *If f is balanced up to n, then $s(f) \geq 1$.*

Proof

1. Let p be the probability that f takes the value 1 and let f_0^* denote the zero-order Fourier coefficient of f. The thesis follows by applying Theorem 4.5 and Lemma 2.5:

$$\begin{aligned} s(f) &= 4\sum_{w} |w|(f^*(w))^2 > 4\,(n-k) \sum_{w:\ |w|\neq 0} (f^*(w))^2 \\ &= 4\,(n-k)\left(\sum_{w} (f^*(w))^2 - (f_0^*)^2\right) = 4\,(n-k)\ (p-p^2) \\ &= n-k\,. \end{aligned}$$

2. As before, the thesis follows from Theorem 4.5 and Lemma 2.5:

$$\begin{aligned} s(f) &= 4\sum_{w} |w|(f^*(w))^2 \geq 4 \sum_{w:\ |w|\neq 0} (f^*(w))^2 \\ &= 4\left(\sum_{w} (f^*(w))^2 - (f_0^*)^2\right) = 4\ (p-p^2) = 1\,. \end{aligned}$$

□

For a strongly balanced function f, we immediately get $s(f) = \Omega((\log n)^d)$, and this is again an indication that $f \notin AC^0[d]$ (see Lemma 2.14).

More in general, similar results could be proved for functions similar to balanced functions, but defined taking a constant different from 1/2 as the probability that the function takes the value 1.

4.4.2 A Family of Balanced functions

The purpose of this section is to demonstrate that there exist k-balanced Boolean functions that are not $(k-1)$-balanced, i.e. that are not so balanced as parity and its complement. To this aim, we now construct a special family of k-balanced and nondegenerated Boolean functions.

Let $1 < k \leq n$, and let $h_{(0)} : \{0,1\}^k \to \{0,1\}$ be

$$h_{(0)}(w_1, w_2, \ldots, w_k) = \begin{cases} \bigwedge_{i=2}^{k} w_i & \text{if} \quad w_1 = 0 \\ \bigvee_{i=2}^{k} w_i & \text{if} \quad w_1 = 1\,. \end{cases}$$

We recursively compute a function f_k, depending on n variables, as follows. Given $h_{(i)}$, depending on $k+i$ variables, we define $h_{(i+1)}$ as the following function

$$h_{(i+1)}(w_1, w_2, \ldots, w_{k+i+1}) = \begin{cases} h_{(i)}(w_2, w_3, \ldots, w_{k+i+1}) & \text{if} \quad w_1 = 0 \\ \neg h_{(i)}(w_2, w_3, \ldots, w_{k+i+1}) & \text{if} \quad w_1 = 1\,. \end{cases}$$

Finally, we take $f_k = h_{(n-k)}$.

Proposition 4.7 *Let $\alpha \in \{0,1\}^{n-k}$, $\beta \in \{0,1\}^k$, with $\beta \equiv (\beta_1, \beta_2, \ldots, \beta_k)$, and $w = \alpha\beta \in \{0,1\}^n$. The Fourier spectrum of f_k has the following structure:*

$$f_k^*(\alpha\beta) = \begin{cases} \frac{1}{2} & |\alpha| = |\beta| = 0 \\ 0 & |\alpha| < n-k \\ 0 & |\alpha| = n-k, \ |\beta| \text{ even} \\ -\frac{1}{2^{k-1}} & |\alpha| = n-k, \ \beta_1 = 0, \ |\beta| \text{ odd} \\ \frac{1}{2^{k-1}} - \frac{1}{2} & |\alpha| = n-k, \ |\beta| = \beta_1 = 1 \\ \frac{1}{2^{k-1}} & |\alpha| = n-k, \ \beta_1 = 1, \ |\beta| \neq 1 \text{ odd}\,. \end{cases}$$

Proof Let us consider the construction of f_k, and, for any $i \geq 1$, let $\alpha \in \{0,1\}^i$ and $\beta \in \{0,1\}^k$. We first prove, by induction on i, the following claim:

$$h_{(i)}^*(\alpha\beta) = \begin{cases} \frac{1}{2} & |\alpha| = |\beta| = 0 \\ 0 & |\alpha| < i, \ |\beta| > 0 \\ h_{(0)}^*(\beta) - \frac{1}{2} & |\alpha| = i, \ |\beta| = 0 \\ h_{(0)}^*(\beta) & |\alpha| = i, \ |\beta| > 0\,. \end{cases}$$

Base
For $i = 1$, we have

$$h_{(1)} = \begin{pmatrix} h_{(0)} \\ \neg h_{(0)} \end{pmatrix}.$$

By the recursive definition of the Hadamard matrix it follows

$$h^*_{(1)} = \frac{1}{2^{k+1}} H_{k+1} h_{(0)} = \frac{1}{2^{k+1}} \begin{pmatrix} H_k & H_k \\ H_k & -H_k \end{pmatrix} \begin{pmatrix} h_{(0)} \\ \neg h_{(0)} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} h^*_{(0)} + (\neg h_{(0)})^* \\ h^*_{(0)} - (\neg h_{(0)})^* \end{pmatrix}.$$

Then the thesis follows since, for any $\beta \in \{0,1\}^k$, we have

$$(\neg h_{(0)})^*(\beta) = \begin{cases} 1 - h^*_{(0)}(\beta) & |\beta| = 0 \\ -h^*_{(0)}(\beta) & |\beta| > 0\,. \end{cases}$$

Induction step
From the definition of $h_{(i)}$ we have

$$h^*_{(i)} = \frac{1}{2} \begin{pmatrix} h^*_{(i-1)} + (\neg h_{(i-1)})^* \\ h^*_{(i-1)} - (\neg h_{(i-1)})^* \end{pmatrix}.$$

Moreover, for any $\alpha_1 \in \{0,1\}$, $\alpha \in \{0,1\}^{i-1}$ and $\beta \in \{0,1\}^k$, by induction hypothesis we have

$$h^*_{(i-1)}(\alpha\beta) = \begin{cases} \frac{1}{2} & |\alpha| = |\beta| = 0 \\ 0 & |\alpha| < i-1 \\ h^*_{(0)}(\beta) - \frac{1}{2} & |\alpha| = i-1,\ |\beta| = 0 \\ h^*_{(0)}(\beta) & |\alpha| = i-1,\ |\beta| > 0\,. \end{cases}$$

Thus, since

$$(\neg h_{(i-1)})^*(\alpha\beta) = \begin{cases} 1 - h^*_{(i-1)}(\alpha\beta) & |\alpha| = |\beta| = 0 \\ -h^*_{(0)}(\alpha\beta) & \text{otherwise}\,, \end{cases}$$

we get

$$h^*_{(i)}(\alpha_1\alpha\beta) = \begin{cases} \frac{1}{2} & \alpha_1 = 0,\ |\alpha| = |\beta| = 0 \\ 0 & \alpha_1 = 0,\ |\alpha| \le i-1,\ |\beta| > 0 \\ 0 & \alpha_1 = 1,\ |\alpha| \le i-1,\ |\beta| \ge 0 \\ h^*_{(i)}(\beta) - \frac{1}{2} & \alpha_1 = 1,\ |\alpha| = i-1,\ |\beta| = 0 \\ h^*_{(i)}(\beta) & \alpha_1 = 1,\ |\alpha| = i-1,\ |\beta| > 0\,, \end{cases}$$

which completes the proof of the claim.

Then, the theorem follows since $f_k \equiv h_{(n-k)}$ and since for $\beta \equiv (\beta_1, \beta_2, \dots, \beta_k) \in \{0,1\}^n$ we have

$$h^*_{(0)}(\beta) = \begin{cases} \frac{1}{2} & |\beta| = 0 \\ 0 & |\beta| \neq 0 \text{ even} \\ -\frac{1}{2^{k-1}} & \beta_1 = 0,\ |\beta| \text{ odd} \\ \frac{1}{2^{k-1}} - \frac{1}{2} & \beta_1 = |\beta| = 1 \\ \frac{1}{2^{k-1}} & \beta_1 = 1,\ |\beta| \neq 1 \text{ odd}\,. \end{cases}$$

□

To conclude, we show that, for any $1 < k < n$, f_k is k-balanced, but it is not $(k-1)$-balanced. Notice that this observation is very important in order to be sure that all the above construction does not collapse onto the parity case.

Proposition 4.8 *The function f_k is balanced up to k, but it is not $(k-1)$-balanced.*

Proof From Proposition 4.7, it follows that $f^*(w) = 0$ for any string w such that $0 < |w| \leq n-k$, while $f^*_0 = \frac{1}{2}$. Thus, from Theorem 4.5, it follows that f_k is k-balanced. Moreover, since there exist nonzero Fourier coefficients of order $n-k+1$, Theorem 4.5 implies that f_k cannot be $(k-1)$-balanced.

□

Chapter 5

Fourier Coefficients as Eigenvalues of Cayley Graphs

The main purpose of this chapter is to show that the Fourier analysis of Boolean functions can be viewed as a Cayley graph eigenvalue problem. This fact allows us to use tools from algebraic graph theory for investigations related to the Fourier coefficients of Boolean functions. Indeed we import results from algebraic graph theory (see [18] and [28]) to analyze Boolean functions with a few distinct Fourier coefficients.

The chapter is organized as follows. In Section 5.1 we show that the Fourier coefficients of a Boolean function are the eigenvalues of a Cayley graph that can be associated to the function. In Section 5.2 we present some consequences of the characterization of Fourier coefficients as graph eigenvalues, and relate the number of nonzero Fourier coefficients of a function to its degree as a polynomial over Z_2 and over R (throughout this chapter we will denote by Z_2 the two element field $GF(2)$). Finally, in Section 5.3 we use the above results to analyze Boolean functions with a few distinct Fourier coefficients: we show that the fewer the number of distinct coefficients is, the stronger are the algebraic properties of the function, and find a nice interpretation for the well known class of *bent functions* in terms of *strongly regular graphs.*

The results presented in this chapter also appeared in [14].

5.1 The Graph G_f associated to a Boolean function f

In this section we first show how it is possible to associate a Cayley graph G_f to any Boolean function f, and then prove that the spectrum of G_f coincides with the Fourier spectrum of f.

Let us first review the definition of *Cayley graph* and *Cayley set*.

Definition 5.1 *Let Γ be a group with identity element e. Suppose C is a Cayley subset of Γ, that is $e \notin C$ and whenever $g \in C$, then $g^{-1} \in C$. The Cayley graph $G = G(\Gamma, C)$ of Γ with respect to C is the graph whose vertex set is Γ, with two vertices g and h adjacent if $gh^{-1} \in C$.*

We slightly modify this definition by dropping the condition $e \notin C$. This generalization is equivalent to allowing the presence of self-loops in the graph.

Let Z_2 be the two elements field, and let $n = 2^k$ Throughout this chapter, a Boolean function will be defined as $f : Z_2^k \to Z_2$. We will also denote by Ω_f the set of strings $w \in Z_2^k$ such that $f(w) = 1$, i.e. $\Omega_f = \{w \in Z_2^k \mid f(w) = 1\}$.

Observe that Z_2^k is a vector space over Z_2, and that for any $w \in Z_2^k$, $w^{-1} = w$ with respect to the $\oplus$ operation. This implies that any subset of Z_2^k is a Cayley set. Thus, we can associate to f the Cayley graph $G_f = G(Z_2^k, \Omega_f)$ of Z_2^k with respect to the set Ω_f. Note that, the vertex set $V(f)$ of G_f is equal to Z_2^k, while the edge set E_f is defined as follows:

$$E_f = \{(w, u) \in Z_2^k \times Z_2^k \mid w \oplus u \in \Omega_f\} = \{(w, u) \in Z_2^k \times Z_2^k \mid f(w \oplus u) = 1\}.$$

It is easy to verify that the adjacency matrix A_f of such a graph is defined as $(A_f)_{i,j} = f(b(i) \oplus b(j))$. A_f is an $n \times n$ symmetric matrix, fully determined by its first row. Moreover, A_f has the following *dyadic property* [5]: $(A_f)_{i,j} = (A_f)_{i+2^{k-1}, j+2^{k-1}}$, $0 \leq i, j < 2^{k-1}$. Finally, it follows from the construction that the rowsums of A_f are all equal to $|\Omega_f|$, and this implies that G_f is a regular graph of degree $|\Omega_f|$.

We turn now our attention to the structure of the graph G_f, and study the properties of its connected components.

Let $\langle \Omega_f \rangle \subseteq Z_2^k$ be the space of the $(0, 1)$ sequences generated by Ω_f. Then, the connected components of the Cayley graph G_f are the cosets $\mu \oplus \langle \Omega_f \rangle$ of $\langle \Omega_f \rangle$ in Z_2^k. Moreover, all the connected components of G_f are isomorphic. Indeed, the map $w \to \mu \oplus w$ induces a bijection of $\langle \Omega_f \rangle$ onto $\mu \oplus \langle \Omega_f \rangle$ which preserves the structure of the components. This can be easily verified since two vertices x, y in G_f are adjacent if and only if $f(x \oplus y) = 1$, and $(\mu \oplus x) \oplus (\mu \oplus y) = x \oplus y$.

Now, let us denote by $dim\langle \Omega_f \rangle$ the dimension of $\langle \Omega_f \rangle$ as a subspace of Z_2^k over Z_2. Since Z_2^k is the disjoint union of its cosets, it follows that G_f has $2^{k - dim\langle \Omega_f \rangle}$ connected components, each of cardinality $2^{dim\langle \Omega_f \rangle}$. Finally, note that whenever $dim\langle \Omega_f \rangle = k$, G_f is connected.

5.1.1 The Spectrum of G_f

The main motivation for introducing the graph G_f is that its spectrum coincides with the Fourier spectrum of its associated Boolean function. This brings the problem of analyzing the Fourier coefficients of Boolean functions into the framework of spectral analysis of graphs, i.e., it makes it possible to use techniques developed to analyze graph spectra for the evaluation of Fourier coefficients.

Following [28], we first review the definition of spectrum of a graph.

Definition 5.2 *Given a graph G and its adjacency matrix A, the eigenvalues of A are called* eigenvalues of G.

Clearly, isomorphic graphs have the same spectrum.

Theorem 5.1 *Let $f : Z_2^k \to Z_2$, and let λ_i, $0 \leq i \leq 2^k - 1$, be the eigenvalues of its associated graph G_f. Then $\lambda_i = 2^k f^*(b(i))$, for any $0 \leq i \leq 2^k - 1$.*

Proof First of all, we know that the eigenvectors of the Cayley graph G_f are exactly the characters of Z_2^k, i.e. the functions $Q_w(x) = (-1)^{w^T x}$. Then λ_i, the i-th eigenvalue of A_f, which corresponds to the eigenvector $Q_{b(i)}$, is given by

$$\lambda_i = \sum_x (-1)^{b(i)^T x} f(x) = 2^k f^*(b(i)) .$$

□

5.2 Consequences

This characterization of Fourier coefficients as graph eigenvalues allows us to immediately state some properties of the Fourier spectrum of a Boolean function. Let $f : Z_2^k \to Z_2$.

Proposition 5.1

1. *The multiplicity of the largest Fourier coefficients of f, $f^*(b(0))$, is equal to $2^{k-dim\langle\Omega_f\rangle}$.*

2. *The number of nonzero Fourier coefficients is given by $rk(A_f)$.*

3. *If G_f is connected, then f has a Fourier coefficient equal to $-\frac{|\Omega_f|}{2^k}$ if and only if its spectrum is symmetric with respect to zero.*

Proof

1. It is a consequence of the fact that the multiplicity of the largest eigenvalue of a regular graph is equal to the number of its connected components (see [28]).

2. It follows from the symmetry of the adjacency matrix A_f.

3. In [28] it is proven that the smallest eigenvalue of a regular connected graph is equal to $-r$, where r is the degree, if and only if the graph is bipartite. Thus, the thesis follows from the fact that a graph containing at least one edge is bipartite if and only if its spectrum, considered as a set of points on the real axis, is symmetric with respect to the zero point.

$\square$

We conclude this section by giving upper and lower bounds on the rank (over the real field) of A_f, i.e. to the number of nonzero Fourier coefficients of the function f, in terms of degrees of polynomials representation of f. We denote by d_2 and d_R the degree of the polynomial representations of f over Z_2 (i.e. over $GF(2)$) and over the real field, respectively.

Lemma 5.1 $2^{d_2} \leq rk(A_f) \leq \sum_{i=1}^{d_R} \binom{k}{i}$.

Proof The degree of f as a real (multi-linear) polynomial equals the maximum order of the nonzero Fourier coefficients (see Definition 2.3). Thus, the upper bound follows since we have $\binom{k}{i}$ coefficients of order i.

In order to prove the lower bound, we first note that for any subfunction f_ρ of f, A_{f_ρ} is a submatrix of A_f, and thus $rk(A_f) \geq rk(A_{f_\rho})$. Then, remember that, because of the structure of the Hadamard matrix H_k, all the Fourier coefficients of a function that takes the value 1 on an odd number of strings, are nonzero (see Proposition 4.2). This implies that, for any f_ρ s.t. $|\Omega_{f_\rho}|$ is an odd integer, A_{f_ρ} has maximum rank. Finally, the bound $rk(A_f) \geq 2^{d_2}$ follows since the degree of a Boolean function f over Z_2 can be characterized as the size of the largest subfunction of f that takes the value 1 on a odd number of strings (see Corollary 4.1).

$\square$

Note that the lower bound is tight for some functions, e.g., the parity function has only two nonzero Fourier coefficients and its degree over Z_2 is equal to 1. Moreover, the upper bound of Lemma 5.1 is always not greater than k^{d_R}. This shows the relation between the degrees of the polynomial representations of f over Z_2 and over the real field and the number of nonzero Fourier coefficients.

5.3 Boolean functions with a few Fourier Coefficients

We say that a Boolean function is *characterized by its spectrum* if it is possible to identify its associated graph (i.e. determine all the details of its topology) only on the basis of the knowledge of its distinct eigenvalues, i.e., without using any information regarding their eigenvectors. Using the results of [28], we will show that functions associated to graphs with one or two distinct eigenvalues are characterized by their spectra. Moreover we will consider the more complex situation of functions related to graphs with three distinct eigenvalues, and show how in some special case these functions are still characterized by their spectra. Note that Boolean functions *characterized by their spectrum* have *low* Kolmogorov complexity because they can be described by a few parameters.

It is interesting to note that the fewer the number of distinct Fourier coefficients is, the stronger are the algebraic properties of the set Ω_f. For instance, it is well known that if a connected graph has exactly m distinct eigenvalues, then its diameter d satisfies $d \leq m - 1$. Thus, if the function f has a constant number of distinct Fourier coefficients, each connected component of G_f has a constant diameter, and this implies that for any $v \notin \Omega_f$, there exist a constant number of strings $w_{(i)} \in \Omega_f$ such that $v = \bigoplus_i w_{(i)}$. We will also see how the well known class of *bent functions* has a nice interpretation in terms of properties of the related graphs.

Since it is very easy to prove that G_f has one eigenvalue if and only if G_f is a totally disconnected graph, i.e. f is the zero constant function, we start our analysis by considering Boolean functions with two distinct Fourier coefficients.

5.3.1 Functions with two distinct Fourier Coefficients

We show that functions whose related graph has only two distinct eigenvalues present a strong algebraic characterization.

Theorem 5.2 *If G_f has two distinct eigenvalues, then its connected components are complete graphs, and $\Omega_f \cup \{b(0)\}$ is a group.*

Proof It is well known that if a graph has exactly m distinct eigenvalues, then its diameter d satisfies the inequality $d \leq m - 1$. In this case, we have $m = 2$, and it follows that $d \leq 1$, i.e. any connected component is a complete graph (note that these complete graphs have self-loops if and only if $b(0) \in \Omega_f$).

Now, observe that for any $u, v \in \Omega_f$, u and v belong to the connected component whose vertex set is $\langle \Omega_f \rangle$. Since the connected components are complete graphs, we have that u and v must be adjacent, and this is equivalent to saying that $f(u \oplus v) = 1$, i.e. $u \oplus v \in \Omega_f$. Finally, since $w \oplus w = b(0)$, it immediately follows that $\Omega_f \cup \{b(0)\}$ is a group.

□

Corollary 5.1

- *If $b(0) \notin \Omega_f$, the two eigenvalues of G_f are $\lambda_0 = |\Omega_f|$ and $\lambda_1 = -1$, with multiplicities $m_0 = 2^k/(|\Omega_f| + 1)$ and $m_1 = 2^k - m_0$, respectively.*

- *If $b(0) \in \Omega_f$, the two eigenvalues of G_f are $\lambda_0 = |\Omega_f|$ and $\lambda_1 = 0$, with multiplicities $m_0 = 2^k/(|\Omega_f|)$ and $m_1 = 2^k - m_0$, respectively.*

Proof The corollary follows from the properties of the graph G_f and from Theorem 5.2.

$\square$

Easy examples of functions in this class are given by the 1-constant function, the OR function and the AND function. Note that the graphs corresponding to the 1-constant function and the OR function are the complete graph with self-loops and without self-loops, respectively. On the contrary, the AND function has an associated graph, with 2^{k-1} connected components, which is simply a matching. These examples correspond, up to isomorphisms, to the cases $dim\langle\Omega_f\rangle = k$ (the constant and the OR functions) and $dim\langle\Omega_f\rangle = 1$ (the AND case). Other functions in this class can be obtained by looking at the intermediate cases in which $1 < dim\langle\Omega_f\rangle < k$.

5.3.2 Functions with three distinct Fourier Coefficients

We now consider Boolean functions with three distinct Fourier coefficients. In order to avoid using a heavy notation, we will make the assumption that G_f is connected. Note that if G_f is not connected, all the results of this section can be restated for the connected component whose vertex set is $\langle\Omega_f\rangle$ by simply substituting Z_2^k with $\langle\Omega_f\rangle$, and with additional care for the other components. Therefore there is no loss of generality in making this assumption.

Graphs with three distinct eigenvalues are known in the literature as *strongly regular* graphs.

Definition 5.3 [28]
A graph $G = G(V, E)$ is called strongly regular*, if there exist nonnegative integers e and d such that for all vertices $u, \in V$ the number of vertices adjacent to both u and v,*

$$\delta(u, v) = |\{w \mid (u, w) \in E \ \wedge \ (v, w) \in E_f\}|$$

is given by

$$\delta(u,v) = \begin{cases} e & \text{if } u \text{ and } v \text{ are adjacent}\,; \\ d & \text{otherwise}\,. \end{cases}$$

Theorem 5.3 [28]
A regular connected graph of degree r is strongly regular if and only if it has exactly three distinct eigenvalues $\lambda_0 = r$, λ_1, λ_2. If G is strongly regular, then $e = r + \lambda_1\lambda_2 + \lambda_1 + \lambda_2$ and $d = r + \lambda_1\lambda_2$.

$\square$

We have the following lemma.

Lemma 5.2 *Let f be a function depending on k binary variables, whose related graph G_f is connected and strongly regular. G_f has diameter 2, and this implies that*
$$\forall\ u \in Z_2^k \setminus \Omega_f\,, \quad \exists\ v \in \Omega_f \ \ s.t.\ \ u \oplus v \in \Omega_f\,.$$

Proof If a graph has exactly m distinct eigenvalues, then its diameter d satisfies the inequality $d \leq m-1$. In this case, we have $m = 3$, and it follows that $d \leq 2$. Moreover, since the only graph with diameter 1 is the complete graph, which has exactly two distinct eigenvalues, the case left is $d = 2$.

We now observe that, by construction, two vertices $w, v \in Z_2^k$ are adjacent in G_f if and only if $w \oplus v \in \Omega_f$. Since the only neighbors of the vertex $b(0)$ are the vertices in Ω_f, we immediately get that any vertices $w \in Z_2^k \setminus \Omega_f$ must be adjacent to a vertex $u \in \Omega_f$, because its distance from $b(0)$ cannot be greater than 2. The thesis then follows because this is equivalent to saying that for any $w \in Z_2^k \setminus \Omega_f$ there must exist a string $u \in \Omega_f$ such that $w \oplus u \in \Omega_f$.

□

Lemma 5.3 *Let G_f be connected and strongly regular. Then $\forall\ v \in Z_2^k$, $\exists\ h$ elements $w \in \Omega_f$ s.t. $v \oplus w \in \Omega_f$, where*

$$h = \begin{cases} e & \text{if} \quad v \in \Omega_f\,; \\ d & \text{if} \quad v \notin \Omega_f\,. \end{cases}$$

Proof First of all, note that for any pair of adjacent vertices u and v, we can give the following characterization to the number e of vertices adjacent to both u and v:

$$e = |(u \oplus \Omega_f) \cap (v \oplus \Omega_f)|\,.$$

This characterization follows from the fact that the set of neighbors of any vertex w is given by $w \oplus \Omega_f$. In the same way, one can easily find that, for any nonadjacent pair of vertices x and y, the number d of vertices adjacent to both x and y is

$$d = |(x \oplus \Omega_f) \cap (y \oplus \Omega_f)|\,.$$

Now, let us consider any string $w \in Z_2^k$ and the string $b(0)$. If $w \in \Omega_f$, then w is adjacent to $b(0)$ and we get $e = |(w \oplus \Omega_f) \cap (b(0) \oplus \Omega_f)| = |(w \oplus \Omega_f) \cap \Omega_f|$, while, if $w \notin \Omega_f$, then w and $b(0)$ are not adjacent and we obtain $d = |(w \oplus \Omega_f) \cap (b(0) \oplus \Omega_f)| = |(w \oplus \Omega_f) \cap \Omega_f|$, and the thesis immediately follows.

□

We can now use the following result.

Proposition 5.2 [28]
The eigenvalues of G_f are

$$\lambda_0 = |\Omega_f| \quad and \quad \lambda_{1,2} = \frac{1}{2}(e - d \pm \sqrt{(e-d)^2 - 4(d - |\Omega_f|)})\,,$$

and their multiplicities are

$$m_0 = 1, \qquad m_1 = \frac{-\lambda_2(2^k - 1) - |\Omega_f|}{\lambda_1 - \lambda_2}, \qquad m_2 = \frac{\lambda_1(2^k - 1) + |\Omega_f|}{\lambda_1 - \lambda_2}.$$

□

Note that, in the case G_f is not connected, the above multiplicities must be multiplied by $2^{k-dim\langle\Omega_f\rangle}$, i.e., the number of components.

If we restate Parseval's identity, which states that $f^*(b(0)) = \sum_{i=0}^{2^k-1} (f^*(b(i)))^2$ (see Section 2.1), in terms of the eigenvalues, Proposition 5.2 yields the following useful equality

$$(r - \lambda_1)(r - \lambda_2) = 2^k(r + \lambda_1\lambda_2), \tag{5.1}$$

where $r = |\Omega_f|$, and where k must be replaced with $dim\langle\Omega_f\rangle$ if G_f is not connected.

Bipartite graphs. Let us analyze the case in which one eigenvalue, say λ_1, is equal to zero and $\lambda_0 = -\lambda_2$. Then, the following theorem shows that G_f is characterized by its spectrum.

Theorem 5.4 [28]
G_f has three distinct eigenvalues $\lambda_0 > \lambda_1 > \lambda_2$, such that $\lambda_1 = 0$ and $\lambda_0 = -\lambda_2$, if and only if G_f is the complete bipartite graph between nodes in Ω_f and nodes in $Z_2^k \setminus \Omega_f$.

□

Note that, under the assumption that G_f is connected, equation (5.1) yields $|\Omega_f| = 2^{k-1}$, and this implies that any function in this class corresponds to the parity of a nonempty subset of the input variables. Indeed, these are the only Boolean functions whose Fourier spectrum is given by the three distinct coefficients $\frac{1}{2}, 0$ and $-\frac{1}{2}$.

Multipartite graphs. We now consider the case in which $\lambda_1 = 0$ and $\lambda_0 + \lambda_2 \neq 0$. Let $r = |\Omega_f|$. For this case, we use the following theorem.

Theorem 5.5 [28]
G_f has eigenvalues r, 0 and λ_2 if and only if its complement is the direct sum of $-\frac{r}{\lambda_2} + 1$ complete graphs of order $-\lambda_2$ (i.e. G_f is a complete multipartite graph*).*

□

Functions belonging to this class are given, e.g. by functions with one subfunction corresponding to a parity function, and with all the other subfunctions (with the same size) identically equal to 1.

Bent functions. Let us now consider the special case in which $e = d$. It is easy to verify that the graphs corresponding to *bent functions* present this property.

Definition 5.4 *Let $f : Z_2^k \to Z_2$, for an even integer k. We say that f is a* bent function *if its Fourier spectrum is such that*

- $f^*(b(0)) = \frac{2^k - 2^{k/2}}{2}$ *or* $\frac{2^k + 2^{k/2}}{2}$*;*
- $|f^*(w)| = \frac{2^{k/2}}{2}$ *for any $w \neq b(0)$.*

See [56] for more information about bent functions.

Lemma 5.4 *Let f be a bent function and G_f its associated graph. Then G_f is strongly regular and has the additional property $e = d$.*

Proof Let us denote by $\lambda_0 > \lambda_1 > \lambda_2$ the eigenvalues of G_f. Then, since $\lambda_1 = -\lambda_2$, the thesis follows immediately from theorem 5.3.

□

We would like to prove that the spectral property $e = d$, which is equivalent to the condition $\lambda_1 + \lambda_2 = 0$, identifies exactly the class of bent functions. In order to prove (or disprove) this claim we should be able to characterize the set of integer solutions of the following equation in r:

$$r^2 - 2^k r + \lambda^2(2^k - 1) = 0\,, \tag{5.2}$$

which is derived from equation (5.1) and where $\lambda = \lambda_1 = -\lambda_2$. If we consider the discriminant Δ of this equation, we get an equality involving only λ:

$$\Delta = 2^{2k} - 4\lambda^2(2^k - 1)\,. \tag{5.3}$$

In order to get an integer solution to (5.2), Δ must be the square of an integer. Thus we immediately get

$$\lambda^2 \leq \frac{2^{2k}}{4(2^k - 1)}\,.$$

We can verify that two integer solutions of (5.2) can be easily found by making the choices $\lambda = 0$ and $\lambda = 1$, which correspond to graphs with only two distinct eigenvalues. On the contrary, a proper solution for the three eigenvalues case can be found my making the choice $\lambda^2 = 2^{k-2}$ which defines exactly the class of bent functions.

In order to find a complete characterization of the class of functions with three distinct nonzero Fourier coefficients and with the additional property $e = d$, we are then left with the problem of understanding whether or not there exist other integer solutions to equation (5.2). Summarizing, we have the following

Proposition 5.3 *Under the conjecture that the integer solutions to equation (5.2) correspond only to the choices $\lambda^2 = 0, 1, 2^{k-2}$, bent functions are the only functions whose associated graph is strongly regular and presents the additional spectral property $e = d$.*

□

We finally observe that, whenever $\lambda_1 = -\lambda_2$, we get $\lambda_1^2 \leq r$. Then, for the class of Boolean functions under study, G_f turns out to be a Ramanujan graph [48]. The class of Ramanujan graphs is indeed defined as the class of connected r-regular graphs whose eigenvalues different from $\pm r$ have absolute value at most $2\sqrt{r-1}$. This class of graphs is very interesting because of its connection with the theory of expander graphs where, in order to get a high expansion, one needs to find graphs whose second largest eigenvalue λ is as small as possible.

Extensions of this analysis seem to be possible. As a first example, for graphs with four distinct eigenvalues, we have a connection, in the bipartite case, with results on balanced incomplete block designs [28].

Chapter 6

Gröbner Basis and Hilbert Function associated with Boolean functions

In the previous chapters we have discussed some mathematical techniques that can be used for Boolean function complexity investigations: harmonic analysis on the hypercube and algebraic graph theory. Here, following Smolensky [65], we analyze the extent to which Boolean function complexity can take advantage of techniques from another mathematical area: algebraic geometry.

Smolensky has observed in [65] that the investigation on Boolean function complexity can be brought into the framework of algebraic geometry. His approach is as follows. A Boolean function f on n variables can be represented as a real multivariate polynomial $p(x_1, , \dots , x_n)$ whose zero set corresponds to the set of zeroes of f, together with the polynomials $x_i^2 - x_i$. Then the affine variety defined by

- $p(x_1, , \dots , x_n) = 0$,
- $x_i^2 - x_i = 0\,,\ \ i = 1, \dots, n$,

describes the zeroes of f, with the proper restrictions on the variables (i.e. $x_i \in \{0, 1\},\ 1 \le i \le n$).

This variety can then be represented in terms of Gröbner bases and its dimension investigated in terms of the Hilbert function, whose value can be used for complexity purposes. More precisely, in [65] Smolensky shows that the values taken by the Hilbert function are related to certain lower bounds, that follow from the comparison of the dimension of the above variety with the dimension of the variety defined by a low degree polynomial.

In this chapter, we provide a complete characterization of the Hilbert function of the ideals associated to symmetric Boolean functions. We then apply this characterization to analyze the behavior of the majority function on symmetric subsets of the Boolean cube, and to derive some nontrivial structural properties of the Hilbert function itself. We believe that these results can give additional insight about the limit of the information content of the Hilbert function and thus about the extent to which the investigation of Boolean function complexity can take advantage of algebraic geometry tools. We also discuss the algebraic mapping from general Boolean functions to monotone ones (noticed in [65]), based on the Gröbner bases associated to the zero sets.

The chapter is organized as follows. In Section 6.1 we recall the basic definitions about Hilbert functions, fix some notation and illustrate the algebraic geometric approach to complexity theory with a brief, revisited exposition of Smolensky's results ([65]). We also stress some elementary, yet important, invariance properties of Hilbert functions. In Section 6.2 we give some preliminary results about Gröbner bases of ideals associated to both monotone and nonmonotone Boolean functions and discuss the algebraic mapping from [65].

Section 6.3 is devoted to the presentation of our main results: we determine the values of the Hilbert function of the ideal associated to any symmetric Boolean function. In Section 6.4 we give a geometric interpretation of our results: we discuss the implications they have on the usefulness of the techniques at issue and view them in relation to the algebraic mapping from [65]. In Section 6.5 we use the results obtained for the Hilbert function of symmetric functions to start an analysis of the behavior of majority on symmetric subsets of the Boolean cube.

The results presented in this chapter also appeared in [16].

6.1 Algebraic Geometry and Complexity Theory

Let $S \subset A^n$ be a subset of the n dimensional affine space over a field k, and let $I(S) \subseteq k[x_1, x_2, \ldots, x_n]$ denote the ideal of polynomials that vanish on S. For any nonnegative integer m, let $k[x_1, x_2, \ldots, x_n]_{\leq m}$ denote the k-module of polynomials of total degree $\leq m$ in $k[x_1, x_2, \ldots, x_n]$. Similarly, let $I(S)_{\leq m}$ denote the k-module of polynomials of total degree $\leq m$ vanishing on S. The value of the *Hilbert function* computed at m, $H_{I(S)}(m)$, is the dimension as a vector space of the set of polynomials of total degree at most m restricted to S. The space of polynomials restricted to S is the quotient space of the space of all polynomials over k, over the ideal $I(S)$. Indeed two polynomials are distinct in the quotient space if and only if they differ at some point of S. The quotient of the corresponding k-modules is therefore the required set (and a k-module itself). More formally, we have the following definition.

Definition 6.1 *The* Hilbert function *of $I(S)$ is the function on the nonnegative integers m defined as the vector-space dimension of the quotient k-module*

$$H_{I(S)}(m) = dim\ k[x_1, x_2, \ldots, x_n]_{\leq m}/I(S)_{\leq m} \, .$$

(For more details, the reader is encouraged to see [7] and [27].)

With some abuse of terminology we often refer to the Hilbert function of the ideal of polynomials vanishing on a set S as the Hilbert function of the set S (or sometimes even as the Hilbert function of the function f, if S is the zero set of f). To simplify the notation, we also write $H_S(m)$ instead of $H_{I(S)}(m)$ whenever no confusion can arise.

We always assume that k is a field of characteristic zero.

We will be dealing with finite sets of points, in particular subsets of the Boolean cube $\{0,1\}^n$, i.e. zero sets of Boolean functions. Notice that for any finite set S, $H_{I(S)}(m) \leq |S|$ for all $m \geq 0$.

On the Boolean cube each monomial is equivalent to a multilinear one, since $x_i^2 = x_i$ for any $1 \leq i \leq n$. Thus in the following when we talk about monomials, we always intend multilinear monomials.

The algebraic geometry techniques naturally apply to monotone Boolean functions, for which the Hilbert function has a very simple interpretation. Indeed, in [65] it was proven that the m-th value of the Hilbert function of the set S of the zeroes of a monotone function is simply given by the number of points in S with 1 in at most m coordinates. This result is an immediate consequence of the fact that the ideal $I(S)$ turns out to be a monomial ideal (see Definition 6.4), when S is the zero set of a monotone function. From this property it becomes clear that, in the case of monotone functions, the information contained in the Hilbert function consists of their correlation with the set of threshold functions, T_m, $m = 0, \ldots, n$, where

$$T_m(x_1, x_2, \ldots, x_n) = \begin{cases} 0 & \text{if } \sum_i x_i < m \\ 1 & \text{if } \sum_i x_i \geq m\,. \end{cases}$$

Indeed the m-th value of the Hilbert function of a monotone function f can be written as

$$\begin{aligned} H_S(m) &= \sum_{w \in \{0,1\}^n} (1 - f(w))(1 - T_{m+1}(w)) = \\ &= \sum_{j=0}^{m} \binom{n}{j} - 2^n p + \sum_{w \in \{0,1\}^n} f(w) T_{m+1}(w)\,, \quad m = 0, 1, \ldots, n, \end{aligned}$$

where $p = \frac{1}{2^n} \sum_w f(w)$, and we set $T_{n+1} = 0$.

Smolensky used these facts to prove a lower bound for the majority function. Precisely, he proved that majority cannot be well approximated by an $ACC[p^r]$ circuit, where p is a prime (see Section 1.1 for some details on ACC circuits).

Smolensky's lower bound. The lower bound for majority is derived essentially in two basic steps: on one side an upper bound is established on the degree of the polynomials that approximate well circuits of a certain class; on the other side it is shown that polynomials whose degree is so bounded are not sufficient to compute accurately majority.

With respect to the first step, it was shown in the works of Razborov [58] and Smolensky [64] that Boolean functions in the class $ACC[p^r]$, where p is a prime, can be approximated by families of degree $o(\sqrt{n})$ polynomials over any field of characteristic p.

The latter step uses the Hilbert function of the zero set of majority to estimate the approximation error.

Smolensky first showed that low degree subsets of the hypercube are characterized by low values of their Hilbert function.

We define the degree of a subset U of $\{0,1\}^n$, to be the minimal degree of a polynomial p such that U can be characterized as the zero set of p. It is possible to find a necessary condition for U to have degree d in terms of the Hilbert function of its complementary set $S = \{0,1\}^n \setminus U$ [65]: for $m = \frac{n-d-1}{2}$,

$$2H_S(m) - |S| \leq 0\,.$$

On the other hand, if a set U can be approximated by the zeroes of a polynomial of degree d, a lower bound for the error E of such an approximation is given by this expression

$$E \geq 2H_S(m) - |S|\,, \tag{6.1}$$

where S and m are as above.

In particular, Smolensky applied the result to $U = majority^{-1}(1)$ to prove that it cannot be well approximated by polynomials of degree $o(\sqrt{n})$. Indeed taking $d = o(\sqrt{n})$, the inequality (6.1) yields:

$$E \geq 2^{n-1} - o(2^n)\,.$$

We can restate Smolensky's result in more general terms. We do so in the following theorem.

Theorem 6.1 *Let f be a monotone function of n variables. Let $S = f^{-1}(\{0\})$ and $S_{\geq k} = \{(w_1, w_2, \ldots, w_n) \in S \mid \sum_i w_i \geq k\ \}$. Then the set $\hat{S}$ of the nonzeroes of any polynomial of degree d is such that*

$$E = |S - \hat{S}| \geq |S| \left(1 - 2\frac{|S_{\geq \frac{n-d-1}{2}}|}{|S|}\right)\,.$$

Proof The proof follows from the characterization of the Hilbert function of the zero sets of monotone functions and from the inequality (6.1) .

□

Note that the lower bound of Theorem 6.1 simply depends on the relationship between the cardinality of the set of the zeroes of the function restricted to points that have 1 in at least $\frac{n-d-1}{2}$ coordinates and the cardinality of its whole zero set. This means that a *high value* of the $\frac{n-d-1}{2}$-th value of the Hilbert function translates into a *high correlation* with $T_{\frac{n-d-1}{2}}$, thus showing that a function with a high correlation with $T_{\frac{n-d-1}{2}}$ cannot be approximated by degree d polynomials.

Invariance of the Hilbert function. Some elementary observations offer a preliminary generalization of the results from [65]. They give us insight about the power of the mathematical techniques we are studying, stressing some limits of the information contained in the Hilbert function of zero sets of Boolean functions.

Since the Hilbert function of a set depends on the relative positions of the points, any transformation of the space that leaves unchanged the latter preserves the values of the Hilbert function.

Isometries (e.g. rotations, reflections, translations and the identity map) preserve distances and therefore belong to the above class of transformations.

Definition 6.2 *Two Boolean functions are* isometrically equivalent *if they have the same zero set up to isometries.*

This implies that by means of Hilbert functions it is not possible to *distinguish* among functions isometrically equivalent.

As a simple application notice that if f is the negation of a monotone Boolean function g, the value at m of its Hilbert function is the number of points in its zero set S with at least $n-m$ ones in their coordinates. Indeed the set of zeroes of f is mapped by a suitable rotation of the space (that takes the point with all ones in its coordinates to the origin) onto the set S' of zeroes of the monotone function g, so that each point in S with 1 in $n-m$ coordinates is mapped to a point of S' with 1 in m coordinates.

In the special case of symmetric functions, all isometries of their zero sets can be obtained via rotations, as a direct consequence of symmetry. Precisely,

Lemma 6.1 *The set of zeroes of a Boolean function f is the image of the zero set of a symmetric function under some isometry if and only if there exists a selected vertex V of the cube, a nonnegative integer ℓ and nonnegative integers $m_1, \ldots, m_\ell$ such that the zero set $Z(f)$ of f can be described as the subset of the cube whose points have Hamming distance m_i (for each $i = 1, \ldots, \ell$) from V.*

Proof Let some vertex V be the image of the n-tuple of zeroes under any isometry; then the n-tuples with m_i ones in the coordinates are mapped by that isometry to those that have Hamming distance m_i from V. Vice versa, for each vertex V of the cube there is a rotation that takes the n-tuple of zeroes to V. □

In particular, notice that for any function whose zero set is the image of the zero set of a symmetric Boolean function under an isometry of the space, there exists a symmetric Boolean function that has the same Hilbert function.

6.2 Smolensky's Mysterious Mapping.

We believe one should study Gröbner bases of Boolean functions in order to get stronger complexity results because they have more information content than Hilbert functions. (Notice that we use the terminology *Gröbner basis of a function* as a shortcut for *Gröbner basis of the ideal associated to the zero set of a function.*) The "mysterious" algebraic mapping from arbitrary Boolean functions to monotone ones, noticed in [65] is indeed defined via Gröbner bases.

The following definitions and classical results illustrate the mathematical relationship between Hilbert functions and Gröbner bases. Before defining the latter we need to review the notions of *monomial ordering* and *monomial ideal.*

Definition 6.3 *A* monomial ordering *on $k[x_1, x_2, \ldots, x_n]$ is any relation $>$ on the set of monomials x^α, $\alpha \in \mathcal{Z}^n_{\geq 0}$, satisfying:*

- *$>$ is a total ordering on $\mathcal{Z}^n_{\geq 0}$.*
- *If $\alpha > \beta$ and $\gamma \in \mathcal{Z}^n_{\geq 0}$, then $\alpha + \gamma > \beta + \gamma$.*
- *$>$ is a well-ordering on $\mathcal{Z}^n_{\geq 0}$. This means that every nonempty subset of $\mathcal{Z}^n_{\geq 0}$ has a smallest element under $>$.*

As we remarked, we only have to deal with multilinear monomials. Thus the exponent α of any variable is either 0 or 1. Therefore any monomial μ over n variables can be associated to a string $\sigma_\mu \in \{0,1\}^n$ that has a 1 in the j-th coordinate if and only if x_j divides μ. This suggests to work with the usual *lexicographic ordering*: given two monomials μ and ν, $\mu < \nu$ if $\sigma_\mu < \sigma_\nu$ as binary numbers. Notice that it is a monomial ordering.

Definition 6.4 *An ideal $I \subseteq k[x_1, x_2, \ldots, x_n]$ is a* monomial ideal *if there is a subset $A \subseteq \mathcal{Z}^n_{\geq 0}$ (possibly infinite) such that I consists of all polynomials which are finite sums of the form $\sum_{\alpha \in A} h_\alpha x^\alpha$, where $h_\alpha \in k[x_1, x_2, \ldots, x_n]$.*

A classical theorem by Macaulay ([49]) proves that the computation of the Hilbert function of a polynomial ideal I can be reduced to the computation of the Hilbert function of a monomial ideal: the *ideal of leading terms* of I.

Given a monomial ordering (say λ) let the *leading term* of a polynomial $p \in k[x_1, x_2, \ldots, x_n]$ be the term associated to its largest monomial with respect to λ. Then, for any ideal I, we can define its ideal of leading terms as the ideal $\langle LT(I) \rangle$ generated by the set of all leading terms of elements in I:

Definition 6.5 *Let $I \subseteq k[x_1, x_2, \ldots, x_n]$ be an ideal, and let $I \neq 0$.*

- *$LT(I)$ denotes the set of leading terms of elements of I.*
- *$\langle LT(I) \rangle$ denotes the ideal generated by the elements of $LT(I)$.*

$\langle LT(I) \rangle$ is a monomial ideal.

Macaulay proved that:

Theorem 6.2 ([49]) *For any ideal $I \subseteq k[x_1, x_2, \ldots, x_n]$, and for any monomial ordering, the monomial ideal $\langle LT(I) \rangle$ has the same Hilbert function as I.*

□

Theorem 6.2 shows that the leading terms of the polynomials in I play an important role in the computation of the Hilbert function. This brings up an important issue concerning $\langle LT(I) \rangle$. Namely, if we are given a finite generating set for I, say $I = \langle f_1, f_2, \ldots, f_s \rangle$, then $\langle LT(f_1), LT(f_2), \ldots, LT(f_s) \rangle$ and $\langle LT(I) \rangle$ may be *different* ideals. In fact $\langle LT(f_1), LT(f_2), \ldots, LT(f_s) \rangle \subseteq \langle LT(I) \rangle$, but the inclusion can be strict. *Gröbner bases* are generating sets for which this doesn't happen:

Definition 6.6 *A finite generating set $G = \{g_1, g_2, \ldots, g_t\}$ for an ideal I is said to be a* Gröbner basis *if*

$$\langle LT(g_1), LT(g_2), \ldots, LT(g_t) \rangle = \langle LT(I) \rangle .$$

Equivalently, a set $\{g_1, \ldots, g_t\} \subseteq I$ is a Gröbner basis of I if and only if the leading term of any element of I is divisible by one of the $LT(g_i)$'s. Moreover, for finite sets $S \subseteq A^n$, we can apply the following lemma.

Lemma 6.2 ([65]) *Let $S \subseteq A^n$ be a finite set. A set $G = \{g_1, g_2, \ldots, g_t\}$ of polynomials, $G \subseteq I(S)$, is a Gröbner basis for $I(S)$ if and only if the number of monomials that are not divisible by any of the leading terms of polynomials in G equals $|S|$.*

Note that, although a given ideal may have many Gröbner bases, we can single out a special one, which turns out to be unique.

Definition 6.7 *A* reduced Gröbner basis *for an ideal I is a Gröbner basis G for I such that:*

- *Any $p \in G$ is monic (i.e. the coefficient of its leading term is equal to 1).*
- *For all $p \in G$, no monomials of p are divisible by any monomial in $\langle LT(G \setminus \{p\}) \rangle$.*

Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function. We denote by $PI(f)$ the set of its prime implicants. Since f is monotone, any prime implicant is given by a product (conjunction) of positive literals, that is any prime implicant is represented by a monomial of the form $x_{i_1} x_{i_2} \cdots x_{i_k}$. Following Smolensky (see [65]), we get the following characterization for the Gröbner basis of f (throughout we use the monomial ordering defined by the usual lexicographical order).

Lemma 6.3 *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. The set of polynomials*

$$G = PI(f) \cup \{\ x_i^2 - x_i \mid x_i \notin PI(f)\ ,\ i = 1, \ldots, n\ \}$$

is the reduced Gröbner basis of the ideal $I(S)$, where $S = f^{-1}(\{0\})$.

Proof We consider the Boolean cube $\{0,1\}^n$ as the subset of the affine variety A^n, consisting of points with coordinates 0 and 1. This restriction on the variables is expressed by the polynomials $x_i^2 - x_i$, $i = 1, 2, \ldots, n$. We then consider the set $\mathcal{I}$ consisting of monomials associated to the ones of f: if $w = (w_1, \ldots, w_n)$ is such that $f(w) = 1$, then the associated monomial is $\prod_{i:\ w_i=1} x_i$.

By applying Lemma 6.2, one can check that the set $\mathcal{I}$, together with the polynomials $x_i^2 - x_i$, $1 \leq i \leq n$, forms a Gröbner basis for $I(S)$.

Starting from this basis, we now construct the reduced Gröbner basis. First of all, note that the set $\mathcal{I}$ is in the ideal generated by the set $PI(f)$. Thus

$$\langle PI(f) \cup \{x_i^2 - x_i \mid i = 1, \ldots, n\}\rangle = \langle \mathcal{I} \cup \{x_i^2 - x_i \mid i = 1, \ldots, n\}\rangle \,.$$

Finally, since by definition a prime implicant cannot be further reduced, the reduced Gröbner basis can be obtained simply by eliminating those polynomials $x_i^2 - x_i$ such that $x_i \in PI(f)$.

□

Through the ideal of leading terms, a transformation that maps arbitrary Boolean functions to monotone ones can be defined. This transformation, which we will denote by $\mathcal{T}$, preserves the value of the Hilbert function and could be a useful tool for reducing some questions about general Boolean functions to questions about monotone functions. Notice that $\mathcal{T}$ depends on the monomial ordering λ.

Theorem 6.3 ([65]) *For any arbitrary function f, there is a monotone function g with the same number of zeroes $|S(g)| = |S(f)|$ and with the same Hilbert function, i.e.*

$$H_{S(f)}(m) = H_{S(g)}(m), \text{ for all } m \geq 0\,.$$

Proof The polynomials in the reduced Gröbner basis of a nonmonotone function are multilinear (with the exception of the polynomials of the form $x_i^2 - x_i$). Thus, for the set $S(f)$ of zeroes of a nonmonotone function f, the ideal of leading terms is generated by some square-free monomials together with the squares of all variables. Hence, it coincides with the ideal of leading terms constructed for a monotone function, say g, with the same number of zeroes, i.e. $|S(g)| = |S(f)|$. Then, the proof follows from Macaulay's Theorem:

$$H_{I(S(f))}(m) = H_{\langle LT(I(S(f)))\rangle}(m) = H_{\langle LT(I(S(g)))\rangle}(m) = H_{I(S(g))}(m)\,.$$

□

By Lemma 6.3, we can give a characterization of the monotone function g associated, via the transformation $\mathcal{T}$, to a nonmonotone function f.

Proposition 6.1 *The function $g = \mathcal{T}(f)$ is the monotone function whose prime implicants are the initial monomials of the reduced Gröbner basis of f.*

By exploiting the above characterizations, it is possible to verify how the reduced Gröbner basis of a monotone Boolean function is related to the monomials in its standard polynomial representation, while Gröbner bases of nonmonotone functions are related to the weak polynomial representation (see for instance [8] for the definition of standard representation and [65] for that of weak representations).

It is very important to notice that, while two functions are indeed mapped to the same monotone one if their reduced Gröbner bases have the same initial monomials, we cannot reach the same conclusion if we only know that they have the same Hilbert function. This is due to the fact that there exist different monotone functions whose associated ideals have the same Hilbert function. For instance, the monotone functions on four variables defined by the sets of prime implicants $\{x_1x_2,\ x_2x_3,\ x_1x_3x_4\}$ and $\{x_1x_2,\ x_3x_4\}$, respectively, have the same Hilbert function. By contrast, observe that no other monotone function can have the same Hilbert function as a threshold one.

6.3 The Hilbert Function of Symmetric functions

Many of the Boolean functions that are of interest to the computer scientists are symmetric, as for instance majority, $\textsc{Mod}_q$, for $q > 2$, (the functions that accept strings with a number of ones not divisible by q) and parity ($\textsc{Mod}_2$). Besides, thanks to symmetry those functions are more manageable than general ones.

The results we obtain for the Hilbert functions of zero sets of symmetric functions enable us to start the exploration of the behavior of majority on subsets of the Boolean cube. The complexity of such relativizations of majority are open problems that promise to give some better insight on the difficulty of Boolean functions.

The analysis we present in this section is quite intricate but the results have clear geometric interpretations, as they give information on the relative positions and dependencies of points in the Boolean cube. The proofs are quite cumbersome but they are interesting in that they expose nontrivial structural properties of the Hilbert function. They use tools from linear algebra.

For ease of exposition let us agree on some terminology:

Definition 6.8 *We call* level with c ones *the set*

$$S_c = \{w \in \{0,1\}^n |\ |w| = c\}.$$

Definition 6.9 *We call* family of degree δ *the set M_δ of all monomials on n variables of degree δ.*

6.3.1 A Single Level

The following lemma provides the natural basis for the result for general symmetric functions. In itself it is perhaps rather technical but its proof is worth the while. It proves the claim reducing the computation of the Hilbert function of a function f to that of two simpler functions. But this reduction can be applied to the general case only with some caution. We postpone the discussion to Section 6.4, after we have proved that we can apply to the zero sets a certain transformation without affecting the values of the Hilbert function.

Lemma 6.4 *Let f be a symmetric Boolean function of n variables and $S = S_c$ (the level with c ones) be its set of zeroes. Then*

$$H_S(m) = \min\left\{\binom{n}{c}, \binom{n}{m}\right\} \textit{ for all } m \geq 0.$$

Proof For $m \geq c$ the result follows by observing that the multilinear monomials of degree c are independent over S_c, and that, for all m, $H_S(m) \leq \binom{n}{c} = |S_c|$.

We prove the lemma by showing that the set of all monomials of degree m is a basis for the space of polynomials of degree at most m on S_c, for $m < c$.

Assume in the following that $c \leq \lfloor \frac{n}{2} \rfloor$. For $c > \lfloor \frac{n}{2} \rfloor$ the result is immediately obtained from the case $c \leq \lfloor \frac{n}{2} \rfloor$ by observing that that there is an isometry that takes the set S_c to the set S_{n-c}.

We first prove that any monomial ν of degree $\delta < m$ on S_c can be obtained as a linear combination of monomials μ_i of degree m, in which only the multiples of ν have nonzero coefficient:

$$\nu = \frac{1}{\binom{c-\delta}{m-\delta}} \sum_{\nu|\mu_i} \mu_i.$$

Indeed, on all points on which ν vanishes, all its multiples vanish as well; besides, on any point on which ν is nonzero, exactly $\binom{c-\delta}{m-\delta}$ multiples of ν of degree m take value one.

Next we have to prove the independence of the family of degree m over n variables on S_c. Indeed, we claim that

$$\sum_i a_i \mu_i = 0 \tag{6.2}$$

on all points of S_c if and only if all a_i's are zero. Let $A_n^{(c,m)}$ be the matrix of the system obtained evaluating (6.2) on all points of S_c. Notice that the $\binom{n}{m}$ columns correspond to monomials while the $\binom{n}{c}$ rows correspond to points and that $\binom{n}{c} > \binom{n}{m}$. We show that this matrix has maximum rank via a double induction on n and on the pair (c, m).

We rearrange the matrix as follows:

- We move all monomials that are multiples of x_n to the right.
- We move all points whose n-th bit is equal to one to the bottom of the matrix.

The matrix $\tilde{A}_n^{(c,m)}$ thus obtained is lower block triangular:

$$\begin{pmatrix} A_0 & 0 \\ C & A_1 \end{pmatrix}$$

Indeed the $\binom{n-1}{c} \times \binom{n-1}{m-1}$ upper right submatrix is null since it corresponds to all the multiples of x_n of degree m computed on the points on which x_n vanishes.

Matrix A_1 is the matrix of the $\binom{n-1}{m-1}$ multiples of x_n of degree m computed on the $\binom{n-1}{c-1}$ points of S_c whose n-th bit is one. It coincides with the matrix $A_{n-1}^{(c-1,m-1)}$ obtained by computing a linear combination

$$\sum_i a_i \mu_i'$$

of all the monomials μ_i' of degree $m-1$ over $n-1$ variables on the points of the level of $\{0,1\}^{n-1}$ with $c-1$ ones. Notice that $A_{n-1}^{(c-1,m-1)}$ has more rows than columns and therefore by the induction hypothesis has maximum rank.

Matrix A_0 is the matrix of the $\binom{n-1}{m}$ monomials of degree m that do not contain x_n, computed on the $\binom{n-1}{c}$ points of S_c whose n-th bit is zero. It coincides with the matrix $A_{n-1}^{(c,m)}$ obtained by computing a linear combination

$$\sum_i a_i \mu_i'$$

of all the monomials μ_i' of degree m over $n-1$ variables on the points of the level of $\{0,1\}^{n-1}$ with c ones. In this case the matrix might still be rectangular, with more rows than columns, or it might become square. The latter happens if $m = n-1-c$; in this case by invariance under isometries we reduce to the case $m = c$, dealt with in the first part of the proof. If on the other hand it is rectangular, the result is granted by the induction hypothesis.

If A_0 and A_1 have maximum rank, so has $\tilde{A}_n^{(c,m)}$.

The basis of the induction is provided on the one side by the case of a square submatrix, on the other by the situation with $m = 0$.

□

6.3.2 General Case

The analysis of the general case shows how the levels and the families are elementary entities in this framework in the sense that each one enters as a whole in the computation of the Hilbert function. This could be seen already in the case of a zero set made of a single level, but it is much more evident here, where the interplay of different levels can be observed along with the interdependencies of families of monomials.

In a way this is not too surprising since we are dealing precisely with functions that cannot distinguish one point of a level from any other, but it is not straightforward from a geometrical point of view if one thinks of the implications it has on the structure of the cube.

It turns out to be natural to associate levels and families of points when thinking about dependencies of monomials and sets of generators. In this way the general problem is ultimately reduced to the case studied in the previous subsection. The associations partly reflect obvious relations, so that if possible the family of a certain degree δ is associated to the level with δ ones, and in general is aimed at maximizing the number of monomials of each family that are independent over the associated level (the proofs will make clear how and why this is so).

We need some extra tools to express the mentioned associations:

$$lev : \{\delta | M_\delta \subseteq M\} \to \{c | S_c \subseteq S\} \text{ and}$$
$$fam : \{c | S_c \subseteq S\} \to \{\delta | M_\delta \subseteq M\},$$

that associate levels of a symmetric set S to families of a set of monomials M and vice versa.

We are going to give an explicit way to compute the Hilbert function of the zero sets of symmetric Boolean functions. Since the proof is quite involved, let us first illustrate it informally.

We first choose a set of monomials which is in general larger than a basis and prove that indeed it contains a basis. Then we show how to actually extract a basis from it and compute its cardinality.

Let $S = S_{c_1} \bigcup \ldots \bigcup S_{c_k}$ be the set of zeroes of a symmetric Boolean function f, where $0 \leq c_1 < c_2 < \ldots < c_k \leq n$ and S_{c_i} is the level with c_i ones. We use the structure of S as a guideline to choose the mentioned set of monomials M, in the sense that we choose a family of monomials for each level S_c of S. In the process we also keep track of the way we associate levels of points to families of monomials. Let us build the set M as follows, starting from the empty set:

- for each $c_i \leq m$, we add to M the family M_{c_i} of all the monomials of degree c_i and record the fact that the family of degree c_i is associated to the level with c_i ones by setting $lev(c_i) = c_i$;

- then we consider all remaining c_i's: let them be $c_{i_m+1} < c_{i_m+2} < \ldots < c_k$. We take the c_i's in increasing order, and for each j add to M the family of monomials M_δ with the largest $\delta \leq m$, which is not already in it, and set $lev(\delta) = c_i$. Formally:
  Let i_m be the largest index such that $c_{i_m} \leq m$. Let $m_0 = m$ (we need m_0 only for uniformity in the definition of the m_i's) and for $i = 1, \ldots, min\{k - i_m, m - k + i_m + 1\}$, $m_i = \max\{j < m_{i-1} | j \neq c_1, \ldots, c_{i_m}\}$. Then, for each i we add to M the family M_{m_i} and we set $lev(m_i) = c_{i_m+i}$.

This discussion should help understand the statement of the following theorem. Notice that the use of the function fam (which is essentially the inverse of the function lev) is motivated by the association between levels and families used in the proof.

Theorem 6.4 *Let*

$$S_{c_1} \bigcup \ldots \bigcup S_{c_k}$$

be the set of zeroes of a symmetric Boolean function f, where $0 \leq c_1 < c_2 < \ldots < c_k \leq n$ and, for each i, $S_{c_i} = \{w \in \{0,1\}^n | \; |w| = c_i\}$. Let i_m be such that $c_{i_m} \leq m < c_{i_m+1}$. Let $m_0 = m$ and for $i = 1, \ldots, min\{k - i_m, m - k + i_m + 1\}$, $m_i = \max\{j < m_{i-1} | j \neq c_1, \ldots, c_{i_m}\}$ and $fam(c_{i_m+i}) = m_i$. Then

$$H_S(m) = \sum_{i=1}^{i_m} \binom{n}{c_i} + \sum_{i=i_m+1}^{k} \min\left\{\binom{n}{c_i}, \binom{n}{fam(c_i)}\right\}. \tag{6.3}$$

It should be noticed that by the above construction M contains $\min\{k, m+1\}$ families of monomials. Also notice that there exists an $h \leq \min\{k, m+1\}$ such that M can be split into two nonintersecting subsets (each one possibly empty):

- $M_{cons} = M_m \bigcup M_{m-1} \bigcup \ldots \bigcup M_{m-h+1}$
- $M_{low} = M_{c_1} \bigcup M_{c_2} \bigcup \ldots \bigcup M_{c_{k-h}}$

with $c_{k-h} < m - h + 1$.

We now start proving results that will be combined together in the proof of Theorem 6.4.

Proposition 6.2 *The monomials in M generate all polynomials of degree at most m on S.*

Proof The proof is in two steps. The core of it consists in showing that if M is made of consecutive families (i.e. if $M = M_{cons}$), then it has the property stated. But first we have to show how the general case can be reduced to the one mentioned.

Let S_c be the level with less ones in S. Notice that if M is not made of consecutive families, then S_c is associated to the family of degree c, and M doesn't contain families of degree smaller than c.

All monomials in $M \setminus M_c$ vanish on S_c. If those monomials can generate all polynomials of degree at most m on $S \setminus S_c$, then it is easy to see that all M generate all polynomials on S (for any polynomial p, let μ_1 be the linear combination of monomials in M_c that generates p on S_c; let μ_2 be the linear combination of monomials in $M \setminus M_c$ that generates $p - \mu_1$ on $S \setminus S_c$; then $\mu_1 + \mu_2$ generates p over all of S).

Inductively, the statement holds, if it holds for a set $M = M_{cons}$.

Notice that, if $m+1 \leq k$, the result is trivial, because M contains all monomials of degree at most m.

In the case $k < m+1$, it is sufficient to show that all monomials of degree less than $m-k+1$ can be generated (where $m-k+1$ is the degree of the smallest family in M).

Consider a generic monomial ν of degree $\delta < m-k+1$. We claim that it can be written as a linear combination of the monomials in M that are multiple of ν, in which monomials of a same degree have the same coefficient:

$$\nu = a_0 \sum_{\nu | \mu_i^m} \mu_i^m + a_1 \sum_{\nu | \mu_i^{m-1}} \mu_i^{m-1} + \ldots + a_{k-1} \sum_{\nu | \mu_i^{m-k+1}} \mu_i^{m-k+1}. \tag{6.4}$$

Notice that on all points on which ν vanishes, all the monomials it divides vanish as well. On the other hand, on any point of a level with c ones, exactly $\binom{c-\delta}{\gamma-\delta}$ monomials of degree γ, all multiples of ν, take value one. Therefore it is sufficient to pick one point from each one of the levels $S_{c_1}, \ldots, S_{c_k}$ and show that the coefficients of the linear combination (6.4) can be chosen so that (6.4) is one on those points.

This gives rise to a system with k equations in k unknowns. The proof that the matrix $A^{(k)}$ of the system has nonzero determinant is rather technical, and doesn't seem to have a clear geometric content. It is an induction on k, that uses a transformation (based on properties of the binomial coefficients) that doesn't change the rank of the matrix.

It must be proven that the matrix $A^{(k)}$ of the system obtained computing the linear combination (6.4) on points $P_{c_1}, \ldots, P_{c_k}$ ($P_{c_i} \in S_{c_i}$) has nonzero determinant. We do it by induction on k, using a transformation (based on properties of the binomial coefficients) that doesn't change the rank of the matrix.

Arrange the matrix so that the columns correspond in the order, from left to right, to the monomials of degree $m, m-1, \ldots, m-k+1$, as in (6.4). Also order the rows so that they correspond, top to bottom, to points with an increasing number of ones.

The crux is to observe that the element $A^{(k)}_{i,j}$ of the matrix is $\binom{c_i-\delta}{m-j-\delta}$, with the convention that, if $a < b$, $\binom{a}{b} = 0$. Then, for each $j = 1, \ldots, k-1$,

$$A^{(k)}_{i,j} = \frac{c_i - m + j}{m - \delta - j + 1} \cdot A^{(k)}_{i,j+1}.$$

If, for each $j = 1, \ldots, k-1$, we subtract to column j column $j+1$ multiplied by $\frac{c_1-m+j}{m-\delta-j+1}$, the generic element of the transformed matrix $\tilde{A}^{(k)}$ becomes

$$\tilde{A}^{(k)}_{i,j} = \frac{c_i - c_1}{m - \delta - j + 1} \cdot A^{(k)}_{i,j}$$

for all $j = 1, \ldots, k-1$.

Thus, the first $k-1$ elements of the first row are annihilated.

The element $\tilde{A}^{(k)}_{1,k} = A^{(k)}_{1,k}$ is nonzero, as all the elements of the anti-diagonal of $A^{(k)}$. Indeed, by the way M has been built and by the hypothesis that $M = M_{cons}$, $m - k + i < c_i$.

The submatrix consisting of the last $k-1$ rows and the first $k-1$ columns, can be seen as the product of three matrices

$$D^{(k-1)} \cdot A^{(k-1)} \cdot \Delta^{(k-1)}$$

where $D^{(k-1)}$ is a nonzero $(k-1) \times (k-1)$ diagonal matrix whose element $D^{(k-1)}_{i,i}$ is $c_{i+1} - c_1$, $\Delta^{(k-1)}$ is also a nonzero $(k-1) \times (k-1)$ diagonal matrix whose element $\Delta^{(k-1)}_{i,i}$ is $\frac{1}{m-\delta-i+1}$ and $A^{(k-1)}$ is a $k \times k$ matrix analogous to $A^{(k)}$ but written for a system in which monomials of degrees $m, m-1, \ldots, m-k+2$ and levels $S_{c_2}, S_{c_3}, \ldots, S_{c_k}$ have been considered.

By induction the thesis follows, since all the elements of the anti-diagonal of $A^{(k)}$ are nonzero.

□

Having proven that M contains a basis, the next step is determining one such basis.

First of all observe that once again we can dispose of the lower levels along with the families associated to them, and reduce the problem to the case in which M consists of consecutive families. Consider as before the level S_c with the least number of ones in S, associated to the family of degree c. We claim that if $B \subseteq M \setminus M_c$ is a basis for the monomials of degree at most M on $S \setminus S_c$, then $B \bigcup M_c$ is a basis for the monomials of degree at most m over all of S. By the proof of Proposition 6.2, $B \bigcup M_c$ are generators. The independence follows from the independence of the monomials in M_c over S_c (cf. Lemma 6.4) and of the monomials in B over $S \setminus S_c$, and from the fact that all the monomials in B vanish over S_c.

Next let us examine the case of $M = M_{cons}$.

Proposition 6.3 *If* $M = M_{cons} = M_m \bigcup \ldots \bigcup M_{m-k+1}$,

$$H_S(m) = \sum_{i=1}^{k} H_{S_{lev(m-i+1)}}(m-i+1).$$

Proof Consider the system of equations expressing the constraints that the coefficients of a linear combination of monomials of M must obey in order to generate a generic monomial μ. We prove that the rank of the matrix of this system can be computed using an induction on the number of pairs (family, level) of families of monomials and levels associated to them.

Consider the matrix $A^{(k,k)}_{lev(m),m}$ of the system, organized as a block matrix in which each block is relative to a family of monomials on a level of points. Precisely, it consists of $k \times k$ blocks where the block (i,j) is the matrix $A^{(1,1)}_{lev(m-i),m-j}$ of dimension $\binom{n}{lev(m-i)} \times \binom{n}{m-j}$ relative to the family of monomials of degree $m-j$ on the points of the level with $lev(m-i)$ ones.

Since the monomials of degree $d+1$ generate all the polynomials of degree at most $d+1$ over any level that has more ones than their degree, we can subtract to each column relative to a monomial of degree d, for $d < m$, the linear combination of monomials of degree $d+1$ that generates that monomial over the level with $lev(m)$ ones. Let

$$\tilde{A}^{(k,k)}_{lev(m),m} = \begin{pmatrix} A^{(1,1)}_{lev(m),m} & 0 \\ \tilde{A}^{(k-1,1)}_{lev(m-1),m} & \tilde{A}^{(k-1,k-1)}_{lev(m-1),m-1} \end{pmatrix}$$

be the matrix obtained with this transformation. It has the same rank as $A^{(k,k)}_{lev(m),m}$, since each column has been substituted by a linear combination that involves the column itself. The effect that the transformation has on the matrix is to annihilate the upper right submatrix $A^{(1,k-1)}_{lev(m),m-1}$, (precisely the one corresponding to the level with $lev(m)$ ones and to the families of degrees $m-1$ down to $m-k+1$). As for the rest of the matrix, each row has been multiplied by some nonzero factor, depending on the number of ones in the level that the row belongs to; the same holds for the columns, except that the factor now depends on the degree of the family that the column belongs to.

If m is larger than the number of ones in all the levels of S other than $S_{lev(m)}$, then the matrix obtained above is even block-diagonal and therefore its rank equals the sum of the ranks of the two submatrices on the diagonal. The lower right matrix $\tilde{A}^{(k-1,k-1)}_{lev(m-1),m-1}$ is the matrix built as $A^{(k,k)}_{lev(m),m}$ but for the monomials $M \setminus M_m$ over the set $S \setminus S_{lev(m)}$, left and right multiplied by two nonnull block-diagonal matrices. Since the multiplying matrices do not affect the rank of $\tilde{A}^{(k-1,k-1)}_{lev(m-1),m-1}$, the thesis follows in this case.

In the other cases we wish to transform the matrix, again without changing its rank, to a block-diagonal one.

$A^{(1,1)}_{lev(m),m}$ is the submatrix of the family of degree m on the level associated to it. If it is a square matrix or has more rows than columns, it has maximum rank because a family of monomials M_δ are independent over any (single) level with no less ones and no less points than δ. Thus we can select enough rows (as many as the columns of $A^{(1,1)}_{lev(m),m}$) which form a basis for all the vectors of length $\binom{n}{m}$ and subtract to each row of the submatrix

$$\tilde{A}^{(k-1,k)}_{lev(m-1),m} = \begin{pmatrix} \tilde{A}^{(k-1,1)}_{lev(m-1),m} & \tilde{A}^{(k-1,k-1)}_{lev(m-1),m-1} \end{pmatrix}$$

a suitable linear combination of those selected rows (the submatrix

$$\tilde{A}^{(1,k)}_{lev(m),m} = \begin{pmatrix} \tilde{A}^{(1,1)}_{lev(m),m} & 0 \end{pmatrix}$$

remains unchanged). The matrix becomes block-diagonal and its rank doesn't change. Besides, the lower right submatrix $\tilde{A}^{(k-1,k-1)}_{lev(m-1),m-1}$ isn't affected by the transformation because the relevant portions of the rows of $A^{(1,k)}_{lev(m),m}$ involved are zero. Thus the thesis follows also in these cases.

Now suppose that $\tilde{A}^{(1,1)}_{lev(m),m}$ has more columns than rows. Here, though, $m < lev(m)$, all levels of S that have less than m ones can be disregarded because the monomials of degree m vanish on those points and by the construction of the set M no other family has been associated to levels that have more than m, but less than $lev(m)$ ones. Thus we only have to take care of levels that have more ones than $S_{lev(m)}$.

Let S_c be any such level. Let us say that a point P_c of S_c *covers* a point $P_{lev(m)}$ of $S_{lev(m)}$ if each coordinate that is one in $P_{lev(m)}$ is one also in P_c. If a monomial μ of degree m is zero on P_c so it is on all points of $S_{lev(m)}$ covered by P_c. If on the other hand it is nonzero on P_c, it is nonzero on $\binom{c-m}{lev(m)-m}$ points covered by P_c. Since this is true for each monomial of degree m, we can subtract to the row corresponding to P_c the sum of the rows corresponding to the points it covers divided by $\binom{c-m}{lev(m)-m}$. As above the matrix obtained is block-diagonal and the thesis follows.

□

We are now ready to put all the tiles of the puzzle together:

Proof of Theorem 6.4 The proof of Proposition 6.3 and the remark preceding it show that a basis can be built in a modular fashion, focusing on a level at a time. In other terms,

$$H_S(m) = \sum_{i=1}^{k-h} H_{S_{c_i}}(m) + \sum_{i=1}^{h} H_{S_{lev(m-i+1)}}(m-i+1),$$

where the first sum is relative to the monomials in M_{low} and the second to those in M_{cons}. By Lemma 6.4, $\min\left\{\binom{n}{c}, \binom{n}{\delta}\right\}$ monomials of degree δ are a basis for the monomials of degree at most δ over S_c, and all of the $\binom{n}{\delta}$ monomials of degree δ are a basis for the monomials of degree at most m (for any $m \geq \delta$) over S_δ. Therefore the above expression for $H_S(m)$ can be rewritten as:

$$H_S(m) = \sum_{i=1}^{k-h} \binom{n}{c_i} + \sum_{i=1}^{h} \min\left\{\binom{n}{lev(m-i+1)}, \binom{n}{m-i+1}\right\}.$$

Define, for $i \leq i_m$ (recall that i_m is the largest index such that $c_{i_m} \leq m$), $fam(c_i) = c_i$, in agreement with the fact that for those i's $lev(c_i) = c_i$ (we need this for the $c_i \leq m$ such that $S_{c_i} \in M_{cons}$). Then the summands in the second sum can be rewritten considering that each corresponds to a level S_{c_i} with $i > k-h$:

$$H_S(m) = \sum_{i=1}^{k-h} \binom{n}{c_i} + \sum_{i=k-h+1}^{k} \min\left\{\binom{n}{c_i}, \binom{n}{fam(c_i)}\right\}.$$

For all c_i in the second sum, such that $fam(c_i) = c_i$, the two binomials clearly coincide; for all $c_i \leq i_m$ this is indeed the case, and thus the relative summands can be moved to the first sum to obtain:

$$H_S(m) = \sum_{i=1}^{i_m} \binom{n}{c_i} + \sum_{i=i_m+1}^{k} \min\left\{\binom{n}{c_i}, \binom{n}{fam(c_i)}\right\},$$

as required.

□

6.4 Consequences

The most immediate geometric consequence of Theorem 6.4 is that the Hilbert function of a set doesn't change if we interchange levels of the same cardinality. The Hilbert function cannot tell whether in the zero set of a function the level with c ones is present or rather the level with $n-c$ ones is.

Corollary 6.1 *Let S_{c_j} be one of the levels of the symmetric set S and suppose that $S_{n-c_j} \not\subset S$. Let $S' = (S \setminus S_{c_j}) \bigcup S_{n-c_j}$. For every $m \geq 0$,*

$$H_{S'}(m) = H_S(m).$$

Proof Consider the set of monomials M and the associations level-family established for the set S. The different situations that can arise are

1. both c_j and $n-c_j$ are not greater than m;
2. both c_j and $n-c_j$ are greater than m;
3. $n-c_j = fam(c_h)$ and $c_h > c_j$;
4. $n-c_j = fam(c_h)$ and $c_h < c_j$;
5. $n-c_j = fam(c_j)$;
6. $n-c_j < fam(c_k)$ (which in turn is smaller than $fam(c_i)$ for all $i < k$).

As we switch to the set S', the corresponding set of monomials M' and therefore the level-family associations may change. Let $\mathcal{C}$ be the set of levels that are affected by the change (in the sense that they are no longer associated to the same families). Let $\mathcal{D}$ be the set of families originally associated to the levels in $\mathcal{C}$.

The fundamental property of the sets $\mathcal{C}$ and $\mathcal{D}$ is that in all cases either all the levels in $\mathcal{C}$ are not smaller (in cardinality) than each family of $\mathcal{D}$, or all of them are not larger.

In the first case the thesis follows since numerically this corresponds to substituting in the sums in (6.3) $\binom{n}{n-c_j}$ for $\binom{n}{c_j}$ and vice versa.

In the cases 2-5, the set of monomials M' equals M although the level-family associations change; in particular all the levels in $\mathcal{C}$ will be still associated to families in $\mathcal{D}$ although in a different order. Notice that in these cases all the levels in $\mathcal{C}$ contribute only to the second sum. By the fundamental property noted above, the sum doesn't change.

In the case 6, $n-c_j$ is smaller than m and thus contributes to the first sum, whereas c_j would contribute to the second sum. But in this case all levels in $\mathcal{C}$ are smaller than all families in $\mathcal{D}$ and thus the thesis follows.

□

This invariance of the Hilbert functions by "reflections" of levels, implies for instance that all the functions whose zero set can be taken to the zero set of the same threshold function via reflections of levels, have the same Hilbert function. Notice that the converse is not true: for instance, the two functions of seven variables whose zero sets are respectively $S_1 \bigcup S_3$ and $S_2 \bigcup S_5$ have the same Hilbert function but their zero sets are not equal modulo reflections of levels. On the other hand it is easy to prove that if two symmetric functions have the same Hilbert function, their zero sets must be made of the same number of levels.

The results proven in this section suggest a grading of symmetric functions based on their Hilbert functions. In this grading functions whose zero set is a single level are at one extreme, since their Hilbert functions are the smallest possible. The threshold functions are at the opposite end, as the functions with highest possible values of the Hilbert function; recall that the idea of the Hilbert function as a way to measure the distance (in some sense) of a function from a threshold function was already suggested in Section 6.1.

Last but not least we wish to explore a very interesting aspect suggested by the proof of Lemma 6.4. As was already observed, that proof presents a way to reduce the computation of the Hilbert function of the zero set of a function f to the same task for functions that in some sense are easier to deal with. More precisely,

$$H_{S(f)}(m) = H_{S(f_0)}(m) + H_{S(f_1)}(m-1),$$

where f_0 and f_1 are the function f restricted to the points that have respectively a 0 and a 1 in the last coordinate. The caveat though is that this decomposition can't be used always. In the proof of Lemma 6.4 itself, the reduction was used only as long as the submatrices were rectangular. The case of square matrices corresponds to functions for which the computation of the Hilbert function is trivial, therefore it is acceptable that we don't try to further simplify that case, albeit perhaps mathematically not too satisfactory. When we turn to the general case, the reduction works only for functions whose levels of zeroes are "as low as possible", i.e. have been reflected below $\left\lfloor \frac{n}{2} \right\rfloor$ whenever possible.

Experimental analysis (we used [6]) suggests that the reduction used in the proof of Lemma 6.4, completed with reflections of levels, is the way to build the sets of initial monomials of the Gröbner bases for the zero sets of symmetric functions. Thus we conjecture that the reduction has stronger properties than those we have used in the proof of Lemma 6.4. Of course before embarking in such a study, one should make sure that those sets are invariant under reflections of levels. Again experimental analysis makes us rather confident that this is the case. In view of the richer information content of Gröbner bases these aspects would be quite interesting and might lead to some understanding of the mapping $\mathcal{T}$.

6.5 An Application

We want to study majority on subsets of the cube as an application of Theorem 6.4. Let MAJ be the majority function. Let MAJ_{even} be the function whose set of zeroes is the intersection of the set of zeroes of MAJ and of that of parity (parity is the function that accepts all strings with an odd number of ones). Define in the analogous way MAJ_{odd} (here we use the negation of parity instead of the function itself). We show that for both these functions we can derive, over any field of characteristic zero, the same nonapproximability results derived in [65] for MAJ, using the above machinery.

Proposition 6.4 *The nonzero sets of* MAJ_{odd} *and* MAJ_{even} *are approximated by sets of degree* $o(\sqrt{n})$ *with errors*

$$\begin{aligned} E_{\text{Maj}_{\text{odd}}} &> 2^{n-2} - o(2^n) \\ E_{\text{Maj}_{\text{even}}} &> 2^{n-2} - o(2^n) \end{aligned}$$

Proof The growth of the Hilbert function of both MAJ_{odd} and MAJ_{even} is related to the parity of $\lfloor \frac{n}{2} \rfloor$. The Hilbert functions of MAJ_{odd} for $\lfloor \frac{n}{2} \rfloor$ odd and of MAJ_{even} for $\lfloor \frac{n}{2} \rfloor$ even, are bigger than half of that of majority. This follows from Theorem 6.4.

For $\lfloor \frac{n}{2} \rfloor$ even, at each level with less than $\lfloor \frac{n}{2} \rfloor$ ones in the zero set of MAJ_{even} corresponds a level with larger cardinality in MAJ_{odd}. But if we add to the zero set of MAJ_{odd} the level with $\lfloor \frac{n}{2} \rfloor$ ones, we increase the Hilbert function by a summand $o(2^n)$ and the enriched zero set has clearly (again use Theorem 6.4) a larger Hilbert function than MAJ_{even}. For $\lfloor \frac{n}{2} \rfloor$ odd, the same holds swapping MAJ_{odd} and MAJ_{even}. Thus the Hilbert functions for MAJ_{odd} and MAJ_{even}, for any n, are certainly larger than

$$\frac{1}{2} H_{S(\text{Maj})}(m) - o(2^n).$$

The cardinalities of the sets of zeroes of both functions are not larger than $2^{n-2} + o(2^n)$, as can be verified directly.

The Hilbert function of MAJ for $m = \frac{n - o(\sqrt{n})}{2}$ is

$$H_{S(\text{Maj})}(m) = 2^{n-1} - o(2^n).$$

Using Smolensky's approximation we derive the claimed lower bound for the errors. In the following formula $*$ stands either for "odd" or for "even" and $m = \frac{n-o(\sqrt{n})}{2}$.

$$\begin{aligned} &E_{\text{Maj}*} \geq \\ &\quad 2H_{S(\text{Maj}*)}(m) - |S(\text{MAJ}*)| > \\ &\quad H_{S(\text{Maj})}(m) - o(2^n) - |S(\text{MAJ}*)| = \\ &\quad 2^{n-1} - 2^{n-2} - o(2^n) = \\ &\quad 2^{n-2} - o(2^n). \end{aligned}$$

□

Using Theorem 6.4 the analogous results could be easily proven, for instance, for MAJ restricted to the zero set of MOD_q (the function that rejects all points with a number of ones which is a multiple of q) or to the zero set of the negation of MOD_q.

But it would be more interesting to study MAJ restricted to subsets of the cube of cardinality $o(2^n)$. At a first analysis it seems that if we restrict our attention to symmetric subsets of that cardinality no satisfactory bound can be derived, since the error seems to tend asymptotically to zero. This leaves open the question of whether one should rather look at nonsymmetric subsets of the cube.

Chapter 7

Conclusions

We have presented a complete framework for analyzing the Fourier Transform of Boolean Functions. We have provided some new applications of Fourier Analysis to Circuit Complexity. We have also found combinatorial properties that make Boolean functions "hard" to compute and showed how the Fourier transform could be used as a mathematical tool for the analysis of Boolean function complexity.

With respect to the sensitivity versus block sensitivity question, the result presented in Chapter 3 provides some evidence of the fact that these complexity measures seem to have a different nature. Therefore, we hope our result could represent the starting point of a deeper analysis, that will eventually lead to the solution of the long standing open problem concerning the interplay between these two measures, possibly by taking advantage of the connection between this problem and that of the existence of odd functions with very low maximal sensitivity, discussed in Chapter 4.

We have shown how the Fourier based techniques can be applied to derive new lower bounds on the size complexity of previously unclassified Boolean functions. These bounds are related to AC^0 functions, and generalize the well known result in Circuit Complexity stating that constant depth circuits require exponential size to compute the parity function.

We have also provided a first look at the Fourier analysis of Boolean functions carried out as a graph eigenvalue problem. We have obtained only preliminary results, but we believe that this approach is promising.

We have shown that the Fourier coefficients of a Boolean function are the eigenvalues of a Cayley graph that can be associated to the function, and we have used the above characterization to analyze Boolean functions with a few distinct Fourier coefficients. We have verified that the fewer the number of distinct Fourier coefficients is, the stronger are the algebraic properties of the function, and found a nice interpretation for the well known class of *bent functions* in terms of *strongly regular graphs*.

Extensions of this approach seem to be possible. As a first example, for graphs with four distinct eigenvalues, we have a connection, in the bipartite case, with results on balanced incomplete block designs [28].

More in general, we think that the Fourier approach can be further exploited to study the complexity of Boolean functions. Indeed, very little is actually known about the structural properties of the Fourier transform of general Booolean functions.

Further work to be done includes a deeper analysis of the structure and of the properties of the Fourier representation of general Boolean functions; a characterization of the Fourier spectrum of functions computable by NC_1 circuits; and an analysis of the complexity of functions whose Fourier spectrum is sparse. Moreover we would like to explore the possibility of applying the *wavelet transform theory* to the study of Boolean function complexity.

With respect to the algebraic geometry approach to Boolean function complexity, we have exposed some limits of the information content of the Hilbert function as a tool to derive complexity bounds. We have stressed that it would be more informative looking at the Gröbner bases of zero sets of functions. One should perhaps start looking at Gröbner bases for symmetric functions. As suggested in Section 6.4 a possible tool that one might exploit to this end is the reduction used in Lemma 6.4. The reduction yields the initial monomials of the reduced Gröbner basis in a straightforward way, provided that they are invariant under reflections of levels (Gröbner bases are clearly not). A better understanding of Gröbner bases seems to be mandatory in order to shed light over the mapping of general Boolean functions to monotone ones. Indeed we have shown that having the same Hilbert function is a necessary but not sufficient condition for two functions to be mapped to the same monotone one. On the other hand the set of initial monomials of the Gröbner basis obviously contains the necessary information.

Moving in another direction, one might want to look deeper into Hilbert functions determining what their properties are for nonsymmetric Boolean functions. It is not clear whether the techniques we used for the symmetric case could be adapted to work more in general for the nonsymmetric one. Common sense suggests that it cannot be so, but the geometric peculiarities of the Boolean cube that we have discovered so far might reserve some surprises.

Some knowledge on the Hilbert function of nonsymmetric functions could help to obtain interesting lower bounds for the approximability by low degree polynomials of the majority function restricted to subsets of the cube of cardinality $o(2^n)$.

Finally one might be interested in investigating what information is given by Hilbert functions computed over fields with finite characteristic. It is known ([64]), for instance, that in such fields sets computed by $ACC[p^r]$ circuits are well approximated by polynomials of degree $o(\sqrt{n})$. Thus, knowledge on the behavior of the Hilbert function in fields with finite characteristic could be applied to prove lower bounds by exploiting non approximability results like those we proved in Section 6.5.

The Hilbert function of any set S computed in finite characteristic certainly grows at most as the Hilbert function of the same set in zero characteristic (because in the former case more polynomials turn out to be equivalent). The "distance" between levels in the zero set of f becomes relevant if k has finite characteristic (by "distance" between levels S_c and S_d we mean he re $|c-d|$), whereas in the case we have studied what mattered was rather the distance of a level from $\frac{n}{2}$. Among other things, this implies that Hilbert functions are no longer invariant under reflections of levels. Thus in a field with nonzero characteristic, the Hilbert function is sensible to different structural properties of the zero sets.

As a final conclusion, we would like to point out how any attempt to find connections between mathematical properties and complexity, has a strong relevance in the field of Complexity Theory. This is due to the lack of mathematical techniques to prove lower bounds for general models of computation. We believe that the techniques developed in this thesis could represent a step in this direction and, more in general, a step forward in the process of understanding the intriguing mathematics which is behind some complexity problems.

Bibliography

[1] M. AJTAI. $\sum_1^1$*-Formulae on Finite Structures.* Annals of Pure and Applied Logic **24**, pp. 1-48, 1983.

[2] N. ALON, R.B. BOPPANA. *The monotone circuit complexity of Boolean functions.* Combinatorica **7(1)**, pp. 1-22, 1987.

[3] N. ALON, O. GOLDREICH, J. HÅSTAD, R. PERALTA. *Simple Constructions of Almost k-wise Independent Random Variables.* Random Structures & Algorithms **3**, pp. 289-304, 1992.

[4] A.E. ANDREEV. *On a method for obtaining lower bounds for th ecomplexity of individual monotone functions.* Dokl. Akad. Nauk SSSR **282(5)**, pp. 1033-1037, 1985 (in Russian). English translation in Soviet Math. Dokl. **31(3)**, pp. 530-534, 1985.

[5] B. ARAZI. *Some properties of Hadamard Matrices Generated Recursively by Kronecker Products.* Linear Algebra and its Applications **25**, pp. 27-39, 1979.

[6] D. BAYER AND M. STILLMAN. *Macaulay: A system for computation in algebraic geometry and commutative algebra.* 1982-1994. Source and object code available for Unix and Macintosh computers. Contact the authors, or download from math.harvard.edu via anonymous ftp.

[7] T. BECKER, V. WEISPFENNING. *Gröbner Bases: A Computational Approach to Commutative Algebra.* Graduate Texts in Mathematics **141**, 1993.

[8] R. BEIGEL. *The Polynomial Method in Circuit Complexity.* Proc. 8th Structure, pp. 82-95, 1993.

[9] M. BEN OR, N. LINIAL. *Collective Coin Flipping.* In *Advances in Computing Research - Randomness and Computation* **5**, pp. 91-115, JAI Press, 1989.

[10] A. BERNASCONI. *Sensitivity vs. Block Sensitivity (an average-case study).* Information Processing Letters, Vol. **59(3)**, pp. 151-157, 1996.

[11] A. BERNASCONI. *On the Complexity of Balanced Boolean functions.* Third Italian Conference on Algorithms and Complexity, Lecture Notes in Computer Science **1203**, pp. 253-263, 1997.

[12] A. Bernasconi, B. Codenotti. *Sensitivity of Boolean Functions, Abstract Harmonic Analysis, and Circuit Complexity.* Technical Report TR-93-030 International Computer Science Institute, Berkeley, CA. 1993.

[13] A. Bernasconi, B. Codenotti. *Measures of Boolean Function Complexity based on Harmonic Analysis.* Second Italian Conference on Algorithms and Complexity, Lecture Notes in Computer Science **778**, pp. 63-72, 1994.

[14] A. Bernasconi, B. Codenotti. *Fourier Analysis of Boolean Functions as a Graph Eigenvalue Problem.* Technical Report IMC-B4-96-19, Istituto di Matematica Computazionale, Consiglio Nazionale delle Ricerche, Pisa. Submitted for publication, 1996.

[15] A. Bernasconi, B. Codenotti, J. Simon. *On the Fourier Analysis of Boolean Functions.* Technical Report IMC-B4-97-03, Istituto di Matematica Computazionale, Consiglio Nazionale delle Ricerche, Pisa. Submitted for publication, 1997.

[16] A. Bernasconi, L. Egidi. *Algebraic geometry tools for complexity lower bounds: the case of symmetric functions.* Technical Report 42/97, Dipartimento di Informatica, Università di Torino. Submitted for publication, 1997.

[17] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, S. Rudich. *Weakly Learning DNF and Characterizing Statistical Query Learning Using Fourier Analysis.* Proc. 26th STOC, pp. 253-262, 1994.

[18] N. Biggs. *Algebraic Graph Theory.* Cambridge University Press, 1968.

[19] R. B. Boppana. *Amplification of Probabilistic Boolean Formulas.* In *Advances in Computing Research - Randomness and Computation* **5**, pp. 27-46, JAI Press, 1989.

[20] Y. Brandman, A. Orlitsky, J. Hennessy. *A Spectral Lower Bound Technique for the Size of Decision Trees and Two-Level AND/OR Circuits.* IEEE Trans. on Computers **39(2)**, pp. 282-287, 1990.

[21] J. Bruck. *Harmonic Analysis of Polynomial Threshold Functions.* SIAM Journal on Discrete Mathematics **3**, pp. 168-177, 1990.

[22] J. Bruck, R. Smolensky. *Polynomial Threshold Functions, AC^0 Functions, and Spectral Norms.* SIAM Journal on Computing **21(1)**, pp. 33-42, 1992.

[23] N.H. Bshouty, C. Tamon. *On the Fourier spectrum of monotone functions.* Journal of the ACM **43(4)**, pp. 747-770, 1996.

[24] S. Bublitz, U. Schürfeld, I. Wegener. *Properties of Complexity Measures for PRAMs and WRAMs.* Theoretical Computer Science **48**, pp. 53-73, 1986.

[25] P. Clote, E. Kranakis. *Boolean Functions, Invariance Groups, and Parallel Complexity.* SIAM Journal on Computing **20(3)**, pp. 553-590, 1991.

[26] S. Cook, C. Dwork, R. Reischuk. *Upper and Lower Bounds for Parallel Random Access Machines with Simultaneous Writes.* SIAM Journal on Computing **15(1)**, pp. 87-97, 1986.

[27] D. Cox, J. Little, D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer-Verlag New York, 1992.

[28] D.M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of Graphs.* Academic Press New York,1979.

[29] M. Dietzfelbinger, M. Kutyłowski, R. Reischuk. *Exact lower time bounds for computing Boolean functions on CREW PRAMs.* Journal of Computer and System Sciences **48(2)**, pp. 231-254, 1994.

[30] M. Dubiner, U. Zwick. *Amplification and Percolation.* SIAM Journal on Computing **26(1)**, pp. 15-38, 1997.

[31] R. Fagin, M.M. Klawe, N.J. Pippenger, L. Stockmeyer. *Bounded-Depth, Polynomial Size Circuits for Symmetric Functions.* Theoretical Computer Science **36**, pp. 239-250, 1985.

[32] Y. Freund. *An improved boosting algorithm and its implications on learning complexity.* Proc. 5th Annual Workshop on Computational Learning Theory, pp.391-398, 1992.

[33] M. Furst, J. Saxe, M. Sipser. *Parity, circuits, and the polynomial-time hierarchy.* Math. Syst. Theory **17**, pp. 13-27, 1984.

[34] C. Gotsman, N. Linial. *Spectral Properties of threshold functions.* Combinatorica **14**, pp. 35-50, 1994.

[35] J. Håstad. *Computational limitations for small depth circuits.* Ph.D. Dissertation, MIT Press, Cambridge, Mass. 1986.

[36] S.L. Hurst, D.M. Miller, J.C. Muzio. *Spectral Method of Boolean Function Complexity.* Electronics Letters **18(33)**, pp.572-574, 1982.

[37] J. Jackson. *The Harmonic Sieve: A Novel Application of Fourier Analysis to Machine Learning Theory and Practice.* PhD Thesis, Carnagie Mellon University, 1995.

[38] J. Kahn, G. Kalai, N. Linial. *The Influence of Variables on Boolean Functions.* Proc. 29th FOCS, pp. 68-80, 1988.

[39] M.G. Karpovsky. *Finite Orthogonal Series in the Design of Digital Devices.* John Wiley and Son, New York, 1976.

[40] C. Kenyon. *Sensitivity vs. Block Sensitivity of Boolean Functions.* DIMACS Technical Report 90-18, 1990.

[41] E. Koutsoupias. *Improvements on Khrapchenko's Theorem.* Theoretical Computer Science **116**, pp. 399-403, 1993.

[42] E. Kushilevitz, Y. Mansour. *Learning Decision Trees using the Fourier Spectrum.* SIAM Journal on Computing **22(6)** pp. 1331-1348, 1993.

[43] M. Kutiłowski. *Time Complexity of Boolean functions on CREW PRAMs.* SIAM Journal on Computing **20(5)**, pp. 824-833, 1991.

[44] R. J. Lechner. *Harmonic Analysis of Switching Functions.* In *Recent Development in Switching Theory*, Academic Press, pp. 122-229, 1971.

[45] N. Linial, Y. Mansour, N. Nisan. *Constant Depth Circuits, Fourier Transform, and Learnability.* Journal of the ACM **40(3)**. pp. 607-620, 1993.

[46] D. Littlewood. *The Theory of Group Characters and the Matrix Representation of Groups.* Oxford University Press, London, 1940.

[47] L.M. Loomis. *An Introduction to Abstract Harmonic Analysis.* Van Nostrand, Princeton, New Jersey, 1953.

[48] A. Lubotzky, R. Phillips, P. Sarnak. *Ramanujan Graphs.* Combinatorica **8**, pp. 261-277, 1988.

[49] F.S. Macaulay *Some Properties of Enumeration in the Theory of Modular Systems.* Proc. London Math. Soc. **26**, pp. 531-555, 1927.

[50] Y. Mansour. *Learning Boolean Functions via the Fourier Transform.* Theoretical Advances in Neural Computing and Learning, Kluwe Academic Publishers, 1994.

[51] J. Naor, M. Naor. *Small-bias Probability Spaces: Efficient Constructions and Applications.* SIAM Journal on Computing **22**, pp. 838-856, 1993.

[52] N. Nisan. *CREW PRAMs and Decision Trees.* SIAM Journal on Computing **20(6)**, pp. 999-1007, 1991.

[53] N. Nisan, M. Szegedy. *On the Degree of Boolean Functions as Real Polynomials.* Proc. 24th STOC, pp.462-467, 1992.

[54] M. Ogiwara, O. Watanabe. *On Polynomial-time Bounded Truth-table Reducibility of NP sets to Sparse sets.* SIAM Journal on Computing **20(3)**, pp. 471-483, 1991.

[55] R. Paturi. *On the Degree of Polynomials that Approximate Symmetric Boolean Functions.* Proc. 24th STOC, pp.468-474, 1992.

[56] O.S. Rothaus. *On "Bent" Functions.* Journal of Combinatorial Theory (A) **20**, pp. 300-305, 1976.

[57] A.A. Razborov. *Lower bounds for the Monotone Complexity of some Boolean Functions.* Dokl. Akad. Nauk SSSR **281**, pp. 798-801, 1985 (in Russian). English translation in Soviet Math. Dokl. **31**, pp. 354-357, 1985.

[58] A.A. Razborov. *Lower bounds for the size of circuits of bounded depth with basis AND, XOR.* Math. Zametski **41**, pp. 598-607, 1987 (in Russian). English translation in Math. Notes **41**, pp. 333-338, 1987.

[59] A.A. Razborov. *On Submodular Complexity Measures.* In "Boolean Function Complexity", Edited by M.S. Paterson, Cambridge University Press, pp.129-139, 1992.

[60] D. Rubinstein. *Sensitivity vs. Block Sensitivity of Boolean Functions.* Combinatorica **15**, 1995.

[61] F. Sellers, M.Y. Hsiao, L.W. Bearnson. *Error Detecting Logic for Digital Computers.* McGraw Hill, 1968.

[62] H .U. Simon. *A tight* $\Omega(\log\log n)$ *bound on the time for parallel RAM's to compute nondegenerate Boolean functions.* FCT'83, Lecture Notes in Computer Science **158**, 1983.

[63] S. Skyum, L.G. Valiant. *A Complexity Theory Based on Boolean Algebra.* Journal of the ACM **32(2)**, pp. 484-502 (1985).

[64] R. Smolensky. *Algebraic Methods in the theory of lower bounds for Boolean circuit complexity.* Proc. 19th STOC, pp. 77-82, 1987.

[65] R. Smolensky. *On Representations by Low-Degree Polynomials.* Proc. 34th FOCS, pp. 130-138, 1993.

[66] A. Subramanian. *The Computational Complexity of the Circuit Value and Network Stability Problems.* PhD Thesis, Tech. Report No. STAN-CS-90-1311, Stanford University, 1990.

[67] M. Szegedy. *Algebraic methods in lower bounds for computational models with limited communication.* Ph.D. Thesis, University of Chicago, 1989.

[68] G. Tardos. *Query complexity, or why is it difficult to separate* $NP^A \cap coNP^A$ *from* P^A *by Random Oracles A ?* Combinatorica **9**. pp.385-392, 1989.

[69] S.-C. Tsai. *Lower bounds on representing Boolean functions as polynomials in* Z_m. Proc. 8th Structure, pp. 96-101, 1993.

[70] L.G. Valiant. *Completeness classes in algebra.* Proc. 11th STOC, pp. 249-261, 1979.

[71] U. Vazirani. *Randomness, Adversaries and Computation.* PhD Thesis, University of California, Berkeley, 1986.

[72] U. Vishkin, A. Wigderson. *Trade-offs between Depth and Width in Parallel Computation.* SIAM Journal on Computing **14(2)**, pp. 303-314, 1985.

[73] I. Wegener. *The complexity of Boolean functions.* Wiley-Teubner Series in Computer Science, New York – Stuttgart, 1987.

[74] I. Wegener, N. Wurm, S.Z. Yi. *Symmetric Functions in AC^0 can be Computed in Constant Depth with Very Small Size.* In "Boolean Function Complexity", Edited by M.S. Paterson, Cambridge University Press, pp. 129-139, 1992.

[75] A. Yao. *Separating the Polynomial-Time Hierarchy by Oracles.* Proc. 26th FOCS, pp. 1-10, 1985.