Studies in Boolean Function Analysis

A thesis submitted to Indian Statistical Institute
in partial fulfillment of the thesis requirements for the degree of
Doctor of Philosophy in Computer Science

Author:
Aniruddha Biswas

Supervisor:
Prof. Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata,
West Bengal, India - 700 108.

Submitted on 3^{rd} June, 2023.

To my family.

Acknowledgements

I would like to take this opportunity to express my heartfelt gratitude and appreciation to all those who have contributed to the successful completion of my thesis. Their support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, I am deeply indebted to my supervisor, Prof. Palash Sarkar, for his unwavering support, expert knowledge, and invaluable guidance. His mentorship has played a crucial role in shaping this research work.

I extend my sincere thanks to all the faculty members of the Indian Statistical Institute (ISI), particularly those from the Applied Statistics Unit (ASU), who have provided assistance and support at various stages of my research. Additionally, I would like to express my appreciation to the staff of the ASU office for their prompt and efficient provision of facilities whenever needed.

The support provided by the Computer and Statistical Services Center (CSSC) of ISI has been invaluable to the success of this research. Their technical expertise and resources have greatly facilitated the implementation and analysis of my work. I would also like to acknowledge the medical unit of ISI for their assistance and care during times of need.

I would like to express my sincere gratitude to my lab mates, Singha-da (Subhadip), Butu-da (Subhabrata), Sanjay-da, Kaushik-da, Sebati-di, Sreyoshi, and Madhurima, for their diverse perspectives and insightful conversations that have significantly enriched my research experience. I would also like to thank Amit-da, Ashwin, Diptendu, Jyotirmoy, and Susanta for their enjoyable discussions and intellectual exchanges. Additionally, I extend my gratitude to my friends, Animesh, Anirban, Anup, Arghya, Avishek, Avijit-da, Avik-da, Chandranan, Gourab, Mostaf, Nilanjan da, Nishant, Prabal, Samir, Sourav, Srimanta-da and Suman for their refreshing company throughout my time at ISI. I am also grateful to Anil-da, Anisur-da, Manab, Ananda, Soumya, Utsav-da, Subhodeep, and Chayan for our regular football matches, which provided a much-needed break and helped stimulate new ideas.

I would like to thank the two anonymous reviewers for their appreciations, suggestions, and detailed comments on my thesis that have greatly helped in refining my thesis.

I am deeply grateful to my parents and my brother for their support, love, and encouragement. Their belief in my abilities have been a constant source of motivation throughout this journey.

Last but certainly not the least, I want to extend a special mention and heartfelt thanks to my wife, Bratati, for her constant support and understanding. Her unwavering belief in

me and her encouragement during challenging times have been instrumental in keeping me focused and motivated.

Siddha Biswas, 4th Dec, 2023

Signature of the Research Fellow with date

Abstract

Boolean functions are important in both theoretical computer science and cryptography. Over the past few decades, significant advancements have been made in this area. The Walsh transform, a variant of the Fourier transform applied to the function $(-1)^g$, where g is a Boolean function, is a vital tool for studying Boolean functions in both fields. The Walsh/Fourier coefficients of a Boolean function offer insights into its properties, and many concepts in both areas can be interpreted in terms of these coefficients. Hence, it is reasonable to assume that analyzing Boolean functions from both perspectives is interconnected, and results from one area can be applied to the other to obtain new outcomes or improve established proof techniques. However, surprisingly, the theory of Boolean functions developed almost parallel in these two fields. One of the objective of this thesis is to investigate and establish connections between various concepts of Boolean functions used in theoretical computer science and cryptography. Through our research, we have solved several existing problems, introduced new ones, and obtained results related to these problems. Furthermore, we have developed new concepts in Boolean function analysis and their applications that are pertinent in both theoretical computer science and cryptography. In the course of our research, we have shown a general counterexample to the "Majority is Least Stable" conjecture, which was previously shown only for $n = 5$. We have also proposed the first-ever lower bound for the "Fourier min-entropy/influence conjecture" in this thesis. Additionally, we utilized programming techniques to explore and unveil some intriguing counting results associated with unate functions and Dedekind numbers.

List of Publications

This thesis is based on the following works. They are mentioned here in their order of appearance in the thesis.

1. Aniruddha Biswas and Palash Sarkar, Separation Results for Boolean Function Classes, (2021) Published in **Cryptography and Communications - Discrete Structures, Boolean Functions and Sequences (CCDS)**, https://link.springer.com/article/10.1007/s12095-021-00488-w.

2. Aniruddha Biswas and Palash Sarkar, Influence of a Set of Variables on a Boolean Function, (2023) Published in **SIAM Journal on Discrete Mathematics (SIDMA)** https://epubs.siam.org/doi/full/10.1137/22M1503531.

3. Aniruddha Biswas and Palash Sarkar, A Lower Bound on the Constant in the Fourier Min-Entropy/Influence Conjecture, (2022) accepted in **Electronic Colloquium on Computational Complexity (ECCC)**, https://eccc.weizmann.ac.il/report/2022/180/

4. Aniruddha Biswas and Palash Sarkar, On the "majority is least stable" conjecture, (2023) Published in **Information Processing Letters**, https://www.sciencedirect.com/science/article/pii/S0020019022000527.

5. Aniruddha Biswas and Palash Sarkar, Counting unate and balanced monotone Boolean functions, (2023), (Accepted for presentation at **BFA 2023** and to be invited to a special issue on Boolean Functions and their Applications in the journal **CCDS**). https://arxiv.org/abs/2304.14069

List of Tables

Contents

Notation

$\mathbb{R}$	: The real numbers
$\mathbb{F}_2$	: The finite field of two elements
$\mathbb{F}_2^n$	: The n-dimensional vector space over the finite field $\mathbb{F}_2$
$f, g, \ldots$	: Functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ will be denoted in the usual font
$W_f, W_g, \ldots$	: Denotes the Walsh transform of $f, g, \ldots$ (Equation 2.6, Page 21)
$\mathscr{f}, \mathscr{g}, \ldots$	: Functions from $\{0,1\}^n$ to $\{0,1\}$ will be denoted in the 'mathpzc' font
$W_{\mathscr{f}}, W_{\mathscr{g}}, \ldots$	: Denotes the Walsh transform of $\mathscr{f}, \mathscr{g}, \ldots$
$\psi, \psi_1, \psi_2, \ldots$	: Functions from $\mathbb{F}_2^n$ to $\mathbb{R}$
$\widehat{\psi}, \widehat{\psi_1}, \widehat{\psi_2}, \ldots$	: Denotes the Fourier transform of $\psi, \psi_1, \psi_2, \ldots$ (Equation 2.3, Page 20)
$\mathfrak{f}, \mathfrak{g}, \ldots$	: Functions from $\{-1,1\}^n$ to $\{-1,1\}$ will be denoted in the 'mathfrak' font
$\widehat{\mathfrak{f}}, \widehat{\mathfrak{g}}, \ldots$	: Denotes the Fourier transform of $\mathfrak{f}, \mathfrak{g}, \ldots$
$P_{\widehat{f}}(k)$	: Denotes the probability assigned by the Fourier transform of f to the integer k (Equation 2.9, Page 22).
$[n]$	: $\{1, 2, \ldots, n\}$
$2^{[n]}$	: Denotes the power set of $[n]$
$\overline{T}$	: If $T \subseteq [n]$, denotes $[n] \setminus T$
$X, Y, Z, \ldots$	: Variables are written in upper case
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \ldots$	: Vectors of variables are written in bold upper case
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	: Vectors are written in bold lower case
$(-1)^{\mathbf{x}}$	: Denotes the vector $((-1)^{x_1}, \ldots, (-1)^{x_n})$, where $\mathbf{x} = (x_1, \ldots, x_n)$
$\mathbf{0}_n, \mathbf{1}_n$	: Denote the all-zero and all-one vectors of length n respectively
χ_T	: For $T \subseteq [n]$, denotes the vector in $\mathbb{F}_2^n$ where the i-th coordinate of χ_T is 1 if and only if $i \in T$
$\#\mathcal{S}$	: Denotes the cardinality of the set $\mathcal{S}$
$\mathcal{C}_{\boldsymbol{\alpha}}$	: Function from $\mathbb{F}_2^n$ to $\{-1,1\}$, defined as $\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) = (-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2}$, for every $\boldsymbol{\alpha} \in \mathbb{F}_2^n$
$\mathsf{inf}_f(i)$	: Denotes the influence of the i^{th} variable on f (Equation 2.17, Page 24)
$\mathsf{inf}(f)$	: Denotes the total influence of f (Equation 2.18, Page 25)
$H(f)$	: Denotes the Fourier entropy of f (Equation 2.10, Page 22)
$H_\infty(f)$	: Denotes the Fourier min-entropy of f (Equation 2.11, Page 23)
$\mathbb{E}(f)$	: Denotes the expectation of f where the random variable $\mathbf{x}$ is chosen uniformly from $\mathbb{F}_2^n$
$\mathsf{Var}(f)$	: Denotes the variance of f, $\mathsf{Var}(f) = \mathbb{E}\left(f - \mathbb{E}(f)\right)^2 = \mathbb{E}(f^2) - \mathbb{E}(f)^2$

$\mathbb{E}_{\mathbf{x}\in\mathcal{S}}(f)$	: Denotes the expectation of f where the random variable $\mathbf{x}$ is chosen uniformly from the set $\mathcal{S}$
$\mathbb{E}_{\mathbf{x}\sim\Phi}(f)$	: Denotes the expectation of f where the random variable $\mathbf{x}$ is chosen uniformly from probability distribution with density Φ
$\mathbf{1}_E$	: $0-1$ indicator random variable for event E
$\Pr_{\mathbf{x}\in\mathbb{F}_2^n}[E]$	: Denotes the probability of the event E where the random variable $\mathbf{x}$ is chosen uniformly from $\mathbb{F}_2^n$
$\log x$	: $\log_2 x$
$\ln x$	: $\log_e x$
maj_n	: The majority function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$
Maj_n	: The majority function from $\{-1,1\}^n$ to $\{-1,1\}$

Chapter 1

Introduction

Boolean functions are binary-valued functions of a finite number of binary-valued variables, which are extensively studied in computer science and mathematics. The investigation of Boolean functions has resulted in a better comprehension of complexity theory, learning theory, coding theory, and cryptography. This has been going on for more than 150 years by various researchers.

Boolean equations are mathematical expressions that use Boolean variables and operators to represent logical relationships between these variables. These equations are of great interest in the field of complexity theory. A Boolean equation is consistent if it has a solution; otherwise, it is inconsistent. A CNF (conjunctive normal form) equation is a specific type of Boolean equation that has the form $\Psi(X) = 1$, where Ψ is in CNF. It has been observed that numerous combinatorial problems can be reduced to the solution of Boolean equations (see, for reference, [64, 63, 80, 49]). This observation was given a more precise formulation by Cook, who proved that every decision problem in the class NP (collection of decision problems that can be solved by a non-deterministic machine in polynomial time) can be transformed in polynomial time into an equivalent CNF equation. Cook's theorem provides a formal link between combinatorial problems and Boolean equations. Cook's theorem [47] is often stated and originally proved to show that the Satisfiability (SAT) problem is NP-complete, which means that it belongs to the collection of decision problems in NP, and that all other NP problems can be polynomial time reducible to it. This result remains true even when each clause of the CNF involves at most three literals, which is known as the 3-Satisfiability or 3-SAT problem. The SAT problem is a decision problem that determines whether a given CNF equation is consistent or not.

The $P \neq NP$ hypothesis states that the class of NP-complete problems cannot be solved efficiently by algorithms. To be considered efficient, an algorithm must run in polynomial time in relation to the input's length. However, researchers have sought to determine if it is possible to efficiently compute approximate solutions to these problems and how accurate these approximations can be. These results are referred to as inapproximability or hardness of approximation results and are typically proven under the assumption of $P \neq NP$. The $\mathsf{Max\text{-}3Lin}(\mathbb{F}_2)$ problem is an example of such a problem, which is defined as follows: An

input instance consists of a list of linear equations of the form $a_1X_1 + a_2X_2 + a_3X_3 = b$, where $a_1, a_2, a_3, b \in \mathbb{F}_2$ are constants, and X_1, X_2, X_3 are variables. In this problem, the objective is to find a set of values, known as an 'assignment', that satisfies the maximum number of linear equations. There exists a trivial approximation algorithm that achieves a multiplicative approximation guarantee of 2. The algorithm simply assigns a random value in $\mathbb{F}_2$ to each variable and, on average, satisfies half of the equations. While the optimal assignment may satisfy all (or nearly all) equations, the assignment produced by the algorithm is within a factor of 2 of the optimal assignment. However, a famous result by Håstad [87] shows that for $\epsilon > 0$, being an arbitrarily small constant, given an instance of $\mathsf{Max\text{-}3Lin}(\mathbb{F}_2)$ that has an assignment satisfying $1 - \epsilon$ fraction of the equations, no efficient algorithm can find an assignment that satisfies $\frac{1}{2}+\epsilon$ fraction of the equations unless $P = NP$. These results regarding inapproximability are closely linked with Fourier analysis of Boolean functions on a Boolean hypercube. Fourier analysis plays an indispensable role in establishing excellent (and often tight) bounds for the hardness of approximation. Further insights on these connections can be found in various references [99, 127].

Computational learning theory is a branch of machine learning that uses mathematical frameworks to measure learning tasks and algorithms. Learning Boolean functions is a fundamental problem in computational learning theory. One popular learning model for Boolean functions is the Probably Approximately Correct (PAC) learning model [169]. In this model, the learner is given a sample of labeled examples drawn from an unknown distribution, and the goal is to learn a Boolean function that is correct on new, unseen examples with high probability. A function is said to be PAC-learnable if there exists an algorithm that can learn the function with high probability, given a sufficiently large sample of labeled examples. Several algorithms have been developed for learning Boolean functions within the PAC model. Some of the popular heuristic decision tree learning algorithms are C4.5 and CART [148, 29]. The positive outcomes of efficient decision tree learning in computational learning theory heavily rely on membership queries [109], which provide the learning algorithm with access to the target function, rather than random examples through an oracle. However, the requirement for membership queries considerably restricts the potential application of such algorithms, and they are unlikely to challenge the popularity of top-down decision tree algorithms without novel ideas. Therefore, it is fair to conclude that, despite their other successes, the models of computational learning theory have not yet provided significant insights into the apparent empirical success of programs like C4.5 and CART [96]. Another significant learning outcome in Boolean function analysis is the learning of constant depth circuits, a class of Boolean functions that can be represented by a fixed-

depth circuit. The learning of constant depth circuits [110] involves decomposing the function into its Fourier coefficients, approximating these coefficients, and constructing a hypothesis for the function. In conclusion, the learning of Boolean functions is an important issue in computational learning theory, and the study of this problem has resulted in significant advancements in learning algorithms and techniques.

Coding theory is a field of mathematics that focuses on developing error-correcting codes for transmission over noisy channels. The aim is to create codes that can detect and correct errors that may occur during transmission. When we refer to codes, we mean systematic methods of transforming data for transmission. Error-correcting codes achieve this by adding extra information, known as redundancy, to the original message before transmission. This redundancy provides a way to detect and fix errors when the message is received. Think of it as sending not only the message but also some additional information that helps to reconstruct the original message accurately in case it's damaged along the way. It is possible to interpret any binary unrestricted code with a length equal to 2^n, where n is a positive integer, as a set of Boolean functions. One specific class of codes, known as Reed-Muller codes [143, 125], is defined using Boolean functions. The Reed-Muller code of order k (where k is a value between 0 and n) is composed of all Boolean functions over $\mathbb{F}_2^n$ that have an algebraic degree bounded above by k [34]. To clarify, the 'algebraic degree' of a Boolean function over $\mathbb{F}_2^n$ is a measure of its complexity, specifically, the highest degree monomial present in the function's representation as a multilinear polynomial, which is known as the algebraic normal form. Lower algebraic degree indicates simpler functions, while higher degree implies more complex functions. Despite their parameters not being very good, except for the first-order Reed-Muller code, these codes are still used today because of their linearity and other properties such as local testability, local decodability, and list decodability. These properties make Reed-Muller codes valuable in the design of probabilistically checkable proofs. The second-order Reed-Muller code, for instance, includes the Kerdock code [97], which is a nonlinear code with a minimum distance that is nearly equal to that of the first-order Reed-Muller code of the same length. Furthermore, the Kerdock code has excellent parameters that are provably optimal among all unrestricted codes [34].

Property testing is a technique used in computer science to verify if a combinatorial structure, such as a graph or a Boolean function, satisfies a particular property or is "far" from satisfying it. In the realm of Boolean functions, various important properties, such as linearity, monotonicity, dictatorship, and junta testing, have been extensively researched. These properties have numerous applications in computer science and other related fields.

Property testing is closely related to the theory of learning Boolean functions. The efficiency of property testing algorithms in learning properties of Boolean functions can be leveraged to design efficient algorithms for learning the functions themselves. Moreover, property testing can also play a crucial role in creating efficient error-correcting codes (locally testable codes). In cases where the communication channel is highly noisy, the decoder may receive a heavily distorted message that cannot be accurately decoded. To avoid the expensive decoding process, it would be useful to have a quick test to determine if the received message is significantly different from any codeword. If the message is "far" from any codeword, the decoder can reject it, otherwise, the decoder can proceed with confidence.

In symmetric key cryptography, Boolean functions are used in the design of cryptographic primitives, such as block ciphers, stream ciphers, and hash functions. Symmetric cryptographic systems are built upon two key principles: confusion and diffusion, which were first introduced by Shannon [155]. The objective of confusion is to hide any algebraic structure present in the system, and this is closely related to the "cryptographic complexity" [34] of the Boolean functions used in the system. On the other hand, diffusion refers to the process of distributing the impact of even minor modifications made to the input data or key across all outputs in the system. Numerous attacks have been discovered against known cryptosystems, highlighting the importance of these principles. The attacks on stream ciphers that have been discovered, resulted in the establishment of criteria that cryptographic Boolean functions must satisfy in order to effectively resist attacks [120, 159]. Some of these criteria are primarily related to confusion, while others primarily pertain to diffusion. It is important to note that in some cases, such as linear attacks, attacks may involve both confusion and diffusion. However, when focusing specifically on S-boxes, it is it is one criterion or the other but not both. In particular, the design of cryptographic Boolean functions must take into account various cryptographic characteristics, such as balancedness, high non-linearity, high algebraic degree, and correlation immunity. It is required to consider all of these properties simultaneously. However, achieving optimum values for all these characteristics at the same time is not possible, and therefore trade-offs must be made. To gain insight into the current state of the art in Boolean and vectorial functions for cryptography, one can refer to several books, such as [50, 175, 36].

Therefore, Boolean functions are important in both theoretical computer science, coding theory and cryptography. Over the past few decades, significant advancements have been made in this area. The Walsh transform, a variant of the Fourier transform applied to the function $(-1)^g$, where g is a Boolean function, is a vital tool for studying Boolean functions

in both fields. The Walsh/Fourier coefficients of a Boolean function offer insights into its properties, and many concepts in both areas can be interpreted in terms of these coefficients. Hence, it is reasonable to assume that analyzing Boolean functions from both perspectives is interconnected, and results from one area can be applied to the other to obtain new outcomes or improve established proof techniques. However, surprisingly, the theory of Boolean functions developed almost parallel in these two fields. One of the objective of this thesis is to investigate and establish connections between various concepts of Boolean functions used in theoretical computer science and cryptography. Through our research, we have solved several existing problems, introduced new ones, and obtained results related to these problems. Furthermore, we have developed new concepts in Boolean function analysis and their applications that are pertinent in both theoretical computer science and cryptography. In the course of our research, we have shown a general counterexample to the "Majority is Least Stable" conjecture, which was previously shown only for $n = 5$. We have also proposed the first-ever lower bound for the "Fourier min-entropy/influence conjecture" in this thesis. Additionally, we utilized programming techniques to explore and unveil some intriguing counting results associated with unate functions and Dedekind numbers.

1.1 Thesis Plan

The thesis comprises five papers [18, 22, 19, 21, 20], and each subsequent chapter's summary is as follows: Chapter 2 covers the essential background material required for the subsequent chapters. Chapter 3 presents a concise review of Fourier analysis on Boolean hypercube.

Chapter 4 presents a discussion of our novel technique for demonstrating the separation between different classes of Boolean functions. The inspiration for this topic came from the following problem proposed by Celerier et. al. in their work [39]. 'Monotone' Boolean functions are applied in various fields, including theoretical computer science and voting theory, while bent functions originally introduced by Rothaus in 1976 [145], are defined only for an even number of variables and find widespread use in cryptography due to their desirable properties. Firstly, their 'derivatives' [36] are balanced, which is crucial in preventing differential attacks on block ciphers. Secondly, the Hamming distance between a bent function and the set of affine Boolean functions is maximum. This directly relates to the linear attack [117] on block ciphers. Celerier et al. attempted to define *monotone* Boolean functions in terms of their Cayley graphs, analogous to the characterization of bent functions [15]. In their attempt to characterize *monotone* Boolean functions, Celerier et. al. derived conditions

on these functions that imply they are not bent. However, they were unable to provide a proof and instead conjectured that no even monotone function can be bent for more than two variables. Carlet et al. later proved this conjecture in [37, 35]. In this thesis, an alternative and concise proof of the same conjecture is provided, utilizing the "total influence" concept (see [127]), which is well-defined in the realm of Boolean functions, as a separator parameter between the two classes. Using this technique, several separation results are presented between classes of Boolean functions studied in coding theory and cryptography and those studied in combinatorics and complexity theory.

The concept of influence of a variable on a Boolean function was first introduced by Ben-Or and Linial [12], and since then it has become a key component in the study of Boolean functions in various contexts. Further, the notion of influence has been extended to consider the influence of a set of variables on a function, and there are four different definitions [12, 164, 62, 23, 68] of this extended notion. While these definitions coincide for a single variable, they differ for multiple variables. Thus, the question arises as to which definition is the most appropriate for the influence of a set of variables. In Chapter 5, we present a systematic and comprehensive study of the notion of influence of a set of variables on a Boolean function. Moreover, we propose a definition of influence based on the auto-correlation function, which is a useful tool for analyzing certain cryptographic properties of Boolean functions. We also discuss several occasions where our definition of influence provides additional insights into the behavior of Boolean functions beyond what can be deduced from existing notions of influence. One such occasion is when we introduce new characterizations of resilient and bent functions using the notion of influence. Our results provide a previously unknown bridge between the notion of influence on the one hand and the notions of bent and resilient functions on the other hand. Using our definition of influence, we also provide generalizations of the Poincaré inequality and the edge expansion property with respect to the influence of a set of variables. Moreover, this new definition simplifies a few proofs of known results.

Chapter 6 introduces a new approach to construct Boolean functions, which is utilized to derive the current best lower bound on the universal constant of the "Fourier min-entropy/influence conjecture" (FMEI) proposed by O'Donnell et. al. in 2011 [131]. The Fourier representation/expansion of a function, which is a multilinear polynomial, is a well-known method for expressing any Boolean function in a unique way. "Fourier entropy" and "total influence" are two measures of the concentration of its Fourier coefficients, which correspond to the monomial coefficients in this representation. In 1996, Friedgut and Kalai

proposed the "Fourier Entropy/Influence conjecture" (FEI), which states that the ratio of entropy to influence of a Boolean function is bounded by a universal constant C. Despite numerous attempts over the years, it remains an open problem. In 2011, O'Donnell et al. replaced the Fourier entropy with "Fourier min-entropy" in the FEI conjecture, resulting in the FMEI conjecture. Although FMEI is a weaker version of FEI, it is also an unsolved problem for general Boolean functions. Currently, there is no research on the lower bound of the universal constant for FMEI. In this chapter, we present an explicit function that has a min-entropy/influence ratio of $128/45 \approx 2.8444$. Moreover, we propose a conjecture regarding the upper bounds of the FEI and FMEI constants for symmetric Boolean functions. It is important to note that O'Donnell et. al. has already demonstrated the validity of the FEI conjecture for symmetric Boolean functions at a constant value of 12.04 [131], which consequently proves the validity of the FMEI conjecture. Our experimental findings suggest that the FEI and FMEI constants are 4 and 2, respectively, but we have not yet been able to prove it and thus propose it as a conjecture.

In Chapter 7 we discuss the resolution of the conjecture put forward by Benjamini, Kalai, and Schramm in 1999 [13], known as the "Majority is the least stable" conjecture, regarding the noise stability of Boolean functions. The noise stability measures the correlation between a Boolean function and its noisy version, and was first studied in [13] for linear threshold functions, which are binary functions whose output is based on whether the weighted sum of their inputs exceeds a certain threshold or not. The majority function is a specific type of linear threshold function. The conjecture proposed in [13] suggested that the majority function has the least noise stability among all linear threshold functions. Although a counterexample exists in the literature for $n = 5$ as demonstrated by Jain [89], we were unable to find any counterexample for arbitrary values of n. In this chapter, we present a counterexample that demonstrates the conjecture's falsehood for all odd values of $n \geq 5$. However, for $n = 1$ and 3, the conjecture remains true.

Unate functions and *monotone* functions have been extensively studied in computer science from various perspectives. The literature contains several works on counting n-variable *monotone* Boolean functions (MBFs), and the value is known up to $n = 9$ [161, 53, 46, 172, 14, 174, 60, 90, 85]. However, although unateness is a generalization of monotonicity, to the best of our knowledge, there is no research on counting n-variable *unate* Boolean functions (UBFs) in the literature. Similarly, while counting 'inequivalent' MBFs is available in the literature [163, 138], there is no such research available for UBFs. Additionally, we did not find any work on counting balanced (inequivalent) MBFs in the literature. In Chapter 8,

we address these gaps in counting and attempt to fill some of them. In this chapter, we demonstrate that counting n-variable unate functions can be reduced to counting n-variable monotone functions. We provide counts for n-variable unate functions up to $n = 9$ and use an enumeration approach to determine the count of n-variable balanced monotone functions up to $n = 7$. Additionally, we establish that counting n-variable balanced unate functions is equivalent to counting n-variable balanced monotone functions, and we calculate the count of n-variable balanced unate functions up to $n = 7$. We also enumerate the numbers of equivalence classes for n-variable balanced monotone functions, unate functions, and balanced unate functions up to $n = 6$. Furthermore, we provide the corresponding counts of n-variable non-degenerate functions for each of these sub-classes.

Chapter 2

Preliminaries

The primary objective of this thesis is the analysis of the Boolean functions. The purpose of this chapter is to introduce the basic definitions and some fundamental tools which we will use throughout the rest of the thesis. We suggest that readers who are already familiar with Boolean functions quickly review this chapter to become familiar with the notation. For those who desire a more thorough introduction to Boolean functions and the analytical tools used to study them, we recommend the following materials: [127, 36].

2.1 Boolean functions

Before considering Boolean functions, let us first take a moment to establish some basic facts concerning their domains. The domain of a Boolean function represents the set of all possible inputs that the function can accept. In the context of Boolean functions, each input variable can take on either a 'true' or 'false' value, and the domain encompasses all possible combinations of these values for a given number of variables. For example, the set of all possible inputs for a Boolean function with 2 variables is (false, false), (false, true), (true, false), and (true, true). It is also possible to associate a geometric structure with the domain of a Boolean function, which is generally called the Boolean hypercube. The Boolean hypercube, also known as the Boolean n-cube, is a geometric structure where each vertex represents the inputs of the Boolean function. Two vertices are connected by an edge if and only if the corresponding inputs differ in exactly one bit, which means their Hamming distance is 1.

In our thesis, we adopt a flexible approach towards representing true and false values of input variables. At times, we use the notation $(0, 1)$ to represent (false,true), which can be interpreted either as elements of the field $\mathbb{F}_2$ or simply as Boolean symbols. Alternatively, we may use $(1, -1)$ to represent (false,true), which can be thought of as real numbers. Corresponding to that there are various representations such as $\mathbb{F}_2^n$ or $\{0, 1\}^n$ or $\{-1, 1\}^n$ of the Boolean hypercube depending on the context. Despite these differences, all representations of the hypercube are equivalent. Each of the representations of the Boolean cube offers

unique advantages depending on the context. For instance, the algebraic structure provided by $\mathbb{F}_2^n$ is particularly useful for generalizing the theory to encompass any finite abelian group. In contrast, when working with Boolean functions over $\{0,1\}^n$, we can conceptualize inputs as strings, aligning well with Boolean logic principles. However, there are scenarios where the need for an algebraic structure is unnecessary, and we prefer to work with arbitrary finite sets of order n. In such cases, the $\{-1,1\}^n$ representation becomes more appropriate. This representation allows us to extend results to any general product spaces. Furthermore, as we will explore in this section, the $\{-1,1\}^n$ representation permits us to view Boolean functions as multilinear polynomials of the input variables. This approach provides a natural means to assess their degree, complexity, and facilitates the concept of learning these functions.

While we primarily use the representation using $\mathbb{F}_2^n$, we also make use of the other representations in some chapters. This section serves to discuss Boolean functions in all the representations that we have used in our thesis.

2.1.1 Boolean functions over $\mathbb{F}_2^n$

The term $\mathbb{F}_2$ represents a finite field that contains only two elements, which can be operated on using two operations: addition (represented by $\oplus$) and multiplication (represented by $\times$). The two elements in the field are denoted by the numbers 0 and 1, and the addition operation is the usual integer addition, but with the result being taken modulo 2. The multiplication operation is the usual integer multiplication modulo 2. Sometimes, xy is used to represent the product of x and y.

For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$, the support of $\mathbf{x}$ is denoted by $\mathsf{supp}(\mathbf{x})$ which is the set $\{i : x_i = 1\}$; the weight of $\mathbf{x}$ is denoted by $\mathsf{wt}(\mathbf{x})$ and is equal to $\#\mathsf{supp}(\mathbf{x})$. For $i \in [n]$, $\mathbf{e}_i$ denotes the vector in $\mathbb{F}_2^n$ whose i-th coordinate is 1 and all other coordinates are 0.

For $\mathbf{x} = (x_1, \ldots, x_2), \mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_2^n$, the "inner product mod 2" function $\langle \mathbf{x}, \mathbf{y} \rangle_2$ of $\mathbf{x}$ and $\mathbf{y}$ is defined as follows.

$$\langle \mathbf{x}, \mathbf{y} \rangle_2 = x_1 y_1 \oplus \cdots \oplus x_n y_n.$$

For a subspace E of $\mathbb{F}_2^n$, $E^\perp$ denotes the subspace $\{\mathbf{x} \in \mathbb{F}_2^n : \langle \mathbf{x}, \mathbf{y} \rangle_2 = 0, \text{ for all } \mathbf{y} \in E\}$. For $T \subseteq [n]$, χ_T denotes the vector in $\mathbb{F}_2^n$ where the i-th coordinate of χ_T is 1 if and only if $i \in T$.

In this thesis, most of the Boolean functions that we consider are maps from the vector

space $\mathbb{F}_2^n$ to $\mathbb{F}_2$ So, a Boolean function will look like

$$f : \mathbb{F}_2^n \to \mathbb{F}_2.$$

Results stated in this representation will be somewhat different from, though equivalent to, the results stated in the other representations. Any Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be uniquely written in the following form.

Definition 1 (*Algebraic normal form*) *The algebraic normal form (ANF) of an n-variable Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the unique canonical expression of the form*

$$f(X_1, \ldots, X_n) \;=\; \bigoplus_{\mathbf{u} \in \mathbb{F}_2^n} a_{\mathbf{u}} \mathbf{X}^{\mathbf{u}}, \tag{2.1}$$

where $a_{\mathbf{u}} \in \mathbb{F}_2$ for $\mathbf{u} \in \mathbb{F}_2^n$, $\mathbf{X} = (X_1, \ldots, X_n)$ and for $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$, $\mathbf{X}^{\mathbf{u}} = X_1^{u_1} \cdots X_n^{u_n}$. The degree of this polynomial is called the algebraic degree of f.

An alternative version of the algebraic normal form, known as the numerical normal form (NNF), has demonstrated its utility in characterizing various cryptographic criteria, as evidenced in [36]. The NNF is the representation of functions $\psi : \mathbb{F}_2^n \to \mathbb{R}$, in the quotient ring $\mathbb{R}[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$ (or $\mathbb{Z}[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$). The existence of this representation for every Boolean function can be straightforwardly deduced from the existence of the ANF as follows:

$$f(\mathbf{X}) = \bigoplus_{\mathbf{u} \in \mathbb{F}_2^n} a_{\mathbf{u}} \mathbf{X}^{\mathbf{u}} \iff (-1)^{f(\mathbf{X})} = \prod_{\mathbf{u} \in \mathbb{F}_2^n} (-1)^{a_{\mathbf{u}} \mathbf{X}^{\mathbf{u}}} \iff f(\mathbf{X}) = \frac{1}{2}\left(1 - 2 \cdot (1 - a_{\mathbf{u}} \mathbf{X}^{\mathbf{u}})\right)$$

The support of a Boolean function is the set of input values for which the function evaluates to 1. Therefore, the support of a Boolean function f, denoted by $\mathsf{supp}(f)$, is the set $\{\mathbf{x} : f(\mathbf{x}) = 1\}$. Then the weight of f, denoted by $\mathsf{wt}(f)$ is equal to $\#\mathsf{supp}(f)$. Note that, $\mathbb{E}(f) = \mathsf{wt}(f)/2^n$.

Definition 2 (*Balanced Boolean function*) *Any n-variable Boolean function f is said to be balanced if $\mathsf{wt}(f) = 2^{n-1}$, i.e., $\mathbb{E}(f) = 1/2$.*

Definition 3 (*(n, k) S-box*) *An (n, k) S-box, also known as an (n, k) vectorial Boolean function, is a mapping $\mathscr{G}$ that takes an n-bit input and produces a k-bit output.*

An (n, k) S-box can be expressed as $\mathscr{G}(\mathbf{X}) = (g_1(\mathbf{X}), \ldots, g_k(\mathbf{X}))$, where $g_1, \ldots, g_k$ are Boolean functions with n variables, referred to as the coordinate functions of the S-box. Any linear combination of coordinate functions is called a component function of the S-box. A balanced (n, k) S-box refers to an S-box whose output distribution is uniformly distributed, meaning that every value of $\mathbb{F}_2^k$ occurs an equal number of times, specifically 2^{n-k} occurrences.

2.1.2 Boolean functions over $\{0,1\}^n$

In this context, a Boolean function with n variables takes a binary input string of length n and produces a single binary output (i.e., a bit). Therefore, in "bit representation" a Boolean function can be written as:

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

where $\{0,1\}^n$ represents the set of all possible input strings of length n, and $\{0,1\}$ represents the set of possible output bits.

Given two bits $a, b \in \{\mathsf{0}, \mathsf{1}\}$, let us define the logical binary operators $\mathsf{AND}(\wedge)$ and OR $(\vee)$ and the logical unary operator NOT $(\neg)$ in the standard manner, i.e.

$$a \wedge b = \begin{cases} \mathsf{1} & \text{if a=b=1,} \\ \mathsf{0} & \text{otherwise.} \end{cases}$$
$$a \vee b = \begin{cases} \mathsf{1} & \text{if a=1 or b=1,} \\ \mathsf{0} & \text{otherwise.} \end{cases}$$
$$\neg a = \begin{cases} \mathsf{1} & \text{if } a = 0, \\ \mathsf{0} & \text{otherwise.} \end{cases}$$

One common way of expressing a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is by using a DNF (disjunctive normal form) formula.

Definition 4 (*DNF*)**.** *A DNF formula over Boolean variables $X_1, \ldots, X_n$ is a logical disjunction (i.e.,* OR*) of terms, where each term is a logical conjunction (i.e.,* AND*) of literals. A literal is either a variable X_i or its negation i.e. $\neg X_i$.*

The complexity of a DNF formula is determined by its size and width. The size of a

DNF formula is the number of terms it contains, while the width is the maximum number of literals in any term.

In addition to DNF formulas, there is also a "dual" notion known as CNF (conjunctive normal form) formulas.

Definition 5 (*CNF*). *A CNF formula is a logical conjunction (i.e.,* AND*) of clauses, where each clause is a logical disjunction (i.e.,* OR*) of literals. The size and width of a CNF formula are defined in the same way as for DNFs.*

2.1.3 Boolean functions over $\{-1,1\}^n$

One way to represent a Boolean function is by using a map from $\{-1,1\}^n$ to $\{-1,1\}$, where the underlying idea is that a bit $b \in \{0,1\}$ is mapped to $(-1)^b$. Thus, in "$\pm$ representation" a Boolean function can be expressed as:

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

The correspondence between $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ is the following: $\mathfrak{f}((-1)^{\mathbf{a}}) = (-1)^{f(\mathbf{a})}$, where for $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$, $(-1)^{\mathbf{a}} = ((-1)^{a_1}, \ldots, (-1)^{a_n})$. In the literature, the "$\pm$ representation" is often used for Boolean functions, as it allows them to be thought of as real numbers, and the Fourier expansion of a Boolean function $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ is simply its representation as a real, multilinear polynomial [127]. To illustrate this concept, let us consider a simple example where $n = 2$, and $\mathfrak{f}$ is max_2, representing the maximum function for 2 bits:

$$\mathsf{max}_2(1,1) = 1, \mathsf{max}_2(-1,1) = 1, \mathsf{max}_2(1,-1) = 1, \mathsf{max}_2(-1,-1) = -1.$$

The max_2 function can be represented as a multilinear polynomial in this manner:

$$\mathsf{max}_2(x_1, x_2) = \frac{1}{2} + \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_1x_2.$$

This expression is known as the "Fourier expansion" of max_2. For any Boolean function $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$, there exists a familiar method to find a polynomial that accurately captures the 2^n values assigned by $\mathfrak{f}$ to the points in $\{-1,1\}^n$. For each point $\mathbf{a} = (a_1, \ldots, a_n)$

in $\{-1,1\}^n$, we can define the indicator polynomial $\mathbb{I}_{\mathbf{a}}(\mathbf{x})$ as follows:

$$\mathbb{I}_{\mathbf{a}}(\mathbf{x}) = \left(\frac{1+a_1x_1}{2}\right)\left(\frac{1+a_2x_2}{2}\right)\ldots\left(\frac{1+a_nx_n}{2}\right).$$

This indicator polynomial assumes the value 1 when $\mathbf{x} = \mathbf{a}$ and the value 0 when $\mathbf{x} \in \{-1,1\}^n \setminus \mathbf{a}$. Consequently, we can represent $\mathfrak{f}$ as a polynomial by using these indicator polynomials:

$$\mathfrak{f}(\mathbf{x}) = \sum_{\mathbf{a}\in\{-1,1\}^n} \mathfrak{f}(\mathbf{a})\mathbb{I}_{\mathbf{a}}(\mathbf{x})$$

.

Let us highlight two important observations regarding this interpolation procedure. Firstly, it is applicable not only to Boolean functions with binary outputs but also to more general cases involving real-valued Boolean functions, where $\mathfrak{f} : \{-1,1\}^n \to \mathbb{R}$. Secondly, since we are primarily interested in inputs $\mathbf{x}$ where $x_i = \pm 1$, and in such cases, any factor involving x_i^2 can simply be replaced by 1, the indicator polynomials maintain their multilinear form when expanded and this interpolation method consistently results in a multilinear polynomial. The multilinear polynomial representing the function $\mathfrak{f}$ can potentially have as many as 2^n terms, corresponding to subsets $S \subseteq [n]$. We denote the monomial corresponding to the subset S as $\mathfrak{C}_S(x_1,\ldots,x_n) = \prod_{i\in S} x_i$. To specify its coefficient, we use the notation $\widehat{\mathfrak{f}}(S)$, which represents the coefficient of the monomial $\mathfrak{C}_S$ in the multilinear representation of $\mathfrak{f}$. This multilinear expression that is $\mathfrak{f}(\mathbf{x}) = \sum_{S\subseteq[n]} \widehat{\mathfrak{f}}(S)\mathfrak{C}_S(\mathbf{x})$ is referred to as the Fourier expansion of $\mathfrak{f}$ and the real number $\widehat{\mathfrak{f}}(S)$ is known as the Fourier coefficient of $\mathfrak{f}$ for the subset S.

In Chapter 7, we presented our findings on the counterexample to the "Majority is least stable conjecture." As the conjecture was originally stated in the "$\pm$ representation" we maintained this representation in that chapter. Additionally, in the same chapter, we thoroughly examine the Fourier transform of the Boolean function $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$, which is necessary for our analysis. This will also clarify the concept of why the aforementioned multilinear representation of a function is unique.

Now, for the alternative representation using $\mathbb{F}_2^n$ or $\{0,1\}^n$, we can define their Fourier expansion encoding the input bits 0 and 1 with the real numbers -1 and 1 respectively. As mentioned earlier, this mapping essentially means that a bit $b \in \{0,1\}$ is mapped to $(-1)^b$. It is important to note that this encoding might not align perfectly with the perspective of

Boolean logic. For example, consider the function max_2 that we discussed earlier, which, in this encoding, represents logical AND. However, despite this interpretation difference, it aligns mathematically with the mapping mentioned above. Furthermore, we can extend the concept of $\mathfrak{C}_S(x_1, \ldots, x_n)$ to two other representations in the following manner: Define $\mathcal{C}_{\boldsymbol{\alpha}} : \mathbb{F}_2^n \to \{-1, 1\}$ as $\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) = (-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2}$. This approach allows us to express the Fourier expansion of any function $\psi : \mathbb{F}_2^n \to \mathbb{R}$ as $\psi(\mathbf{x}) = \sum_{S \subseteq [n]} \widehat{\psi}(S) \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x})$. In fact, if we consider $\mathbb{F}_2^n$ as the n-dimensional vector space over $\mathbb{F}_2$, it becomes meaningful to associate subsets $S \subseteq [n]$ with vectors $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. We will discuss the Fourier transform over $\mathbb{F}_2^n$ in detail in the next section.

2.2 Fourier transform

It is more convenient to study the Fourier transform of Boolean functions by looking at a larger class of functions, which map the Boolean hypercube $\mathbb{F}_2^n$ to the set of real numbers $\mathbb{R}$.

The set of functions from $\mathbb{F}_2^n \to \mathbb{R}$ forms a vector space $\mathcal{V}$ over $\mathbb{R}$ since we can add two functions point wise and can multiply a function by a real scalar. We define the inner product $\langle \psi_1, \psi_2 \rangle$ on pairs of function $\psi_1, \psi_2 : \mathbb{F}_2^n \to \mathbb{R}$ in this vector space by

$$\langle \psi_1, \psi_2 \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \psi_1(\mathbf{x}) \psi_2(\mathbf{x}) = \mathbb{E}(\psi_1 \cdot \psi_2). \tag{2.2}$$

We also use the notation $\|\psi\|_2 = \sqrt{\langle \psi, \psi \rangle}$, and more generally $\|\psi\|_p = \mathbb{E}(|\psi|^p)^{\frac{1}{p}}$ for function $\psi : \mathbb{F}_2^n \to \mathbb{R}$.

To explain the Fourier transform on the Boolean hypercube, it is essential to first present the idea of characters for the Boolean hypercube. For every $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, the function $\mathcal{C}_{\boldsymbol{\alpha}} : \mathbb{F}_2^n \to \{-1, 1\}$ defined by $\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) = (-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2}$ is an example of a character of $\mathbb{F}_2^n$.

Lemma 1 $\mathbb{E}(\mathcal{C}_{\boldsymbol{\alpha}}) = \begin{cases} 1 & \text{if } \boldsymbol{\alpha} = \mathbf{0}_n, \\ 0 & \text{otherwise.} \end{cases}$

Lemma 2 *For* $\mathbf{x} \in \mathbb{F}_2^n$, $\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) \cdot \mathcal{C}_{\boldsymbol{\beta}}(\mathbf{x}) = \mathcal{C}_{\chi_{\mathsf{supp}(\boldsymbol{\alpha}) \triangle \mathsf{supp}(\boldsymbol{\beta})}}$, *where* $\mathsf{supp}(\boldsymbol{\alpha}) \triangle \mathsf{supp}(\boldsymbol{\beta})$ *denotes symmetric difference.*

Proof: For $\mathbf{x} = (x_1, \ldots, x_2), \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{F}_2^n$,

$$\mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) \cdot \mathcal{C}_{\boldsymbol{\beta}}(\mathbf{x}) = (-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2} \cdot (-1)^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle_2} = (-1)^{\oplus_{i \in \mathsf{supp}(\boldsymbol{\alpha}) \triangle \mathsf{supp}(\boldsymbol{\beta})} x_i} = \mathcal{C}_{\chi_{\mathsf{supp}(\boldsymbol{\alpha}) \triangle \mathsf{supp}(\boldsymbol{\beta})}}.$$

□

As a result of equation 2.2, Lemma 1 and Lemma 2, the next theorem can be immediately deduced.

Theorem 1 $\langle \mathcal{C}_{\boldsymbol{\alpha}}, \mathcal{C}_{\boldsymbol{\beta}} \rangle = \begin{cases} 1 & \text{if } \boldsymbol{\alpha} = \boldsymbol{\beta}, \\ 0 & \text{otherwise.} \end{cases}$

With the necessary background information in place, we can now move on to discussing the Fourier transform on Boolean hypercube.

Definition 6 (*Fourier transform*). *Let $\psi : \mathbb{F}_2^n \to \mathbb{R}$. The Fourier transform of ψ is a map $\widehat{\psi} : \mathbb{F}_2^n \to \mathbb{R}$ which is defined as follows.*

$$\widehat{\psi}(\boldsymbol{\alpha}) = \langle \psi, \mathcal{C}_{\boldsymbol{\alpha}} \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \psi(\mathbf{x})(-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2}. \tag{2.3}$$

Definition 7 (*Inverse Fourier transform*). *Given $\widehat{\psi}$, it is possible to recover ψ using the following inverse formula.*

$$\psi(\mathbf{x}) \quad = \quad \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \widehat{\psi}(\boldsymbol{\alpha})(-1)^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2}. \tag{2.4}$$

The above expression is sometimes referred to as the Fourier expansion of ψ, and $\widehat{\psi}(\boldsymbol{\alpha})$ is known as the Fourier coefficient of ψ at $\boldsymbol{\alpha}$.

The vector space $\mathcal{V}$ has dimension 2^n, and functions in it can be considered as vectors in $\mathbb{R}^n$. According to Theorem 1, the 2^n characters $\mathcal{C}_{\boldsymbol{\alpha}}$ form an orthonormal basis for $\mathcal{V}$, meaning that every function $\psi : \mathbb{F}_2^n \to \mathbb{F}_2$ in $\mathcal{V}$ can be expressed as a unique linear combination of these characters. The coefficients of this linear combination are the Fourier coefficients, as shown in equation 2.4.

Given two functions, $\psi_1, \psi_2 : \mathbb{F}_2^n \to \mathbb{R}$, we can calculate their inner product by taking the dot product of their coordinates in the orthonormal basis of characters. This formula is known as Plancherel's theorem.

Theorem 2 (*Plancherel's theorem*) .[*127*] *For any $\psi_1, \psi_2 : \mathbb{F}_2^n \to \mathbb{R}$, $\langle \psi_1, \psi_2 \rangle = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \widehat{\psi_1}(\boldsymbol{\alpha}) \widehat{\psi_2}(\boldsymbol{\alpha})$.*

Proof: Replacing ψ_1 and ψ_2 in (2.2) we obtain the following equation.

$$\langle \psi_1, \psi_2 \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \left(\sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \widehat{\psi_1}(\boldsymbol{\alpha}) \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) \right) \left(\sum_{\boldsymbol{\beta} \in \mathbb{F}_2^n} \widehat{\psi_2}(\boldsymbol{\beta}) \mathcal{C}_{\boldsymbol{\beta}}(\mathbf{x}) \right) = \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{F}_2^n \\ \boldsymbol{\beta} \in \mathbb{F}_2^n}} \widehat{\psi_1}(\boldsymbol{\alpha}) \widehat{\psi_2}(\boldsymbol{\beta}) \langle \mathcal{C}_{\boldsymbol{\alpha}}, \mathcal{C}_{\boldsymbol{\beta}} \rangle.$$

Now, By making use of Theorem 1 we arrive at the desired result. □

Parseval's theorem is a specific instance of Plancherel's theorem. It states that the Fourier transform preserves the squared L_2 norm of a function.

Theorem 3 (*Parseval's theorem*) .[*127*] *For any* $\psi : \mathbb{F}_2^n \to \mathbb{R}$, $\|\psi\|_2^2 = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \left(\widehat{\psi}(\boldsymbol{\alpha}) \right)^2$.

The Poisson summation formula ([36]) provides a useful relation between a function $\psi : \mathbb{F}_2^n \to \mathbb{R}$ and its Fourier transform.

Theorem 4 (*Poisson summation formula*) .[*36*] *Let* E *be a subspace of* $\mathbb{F}_2^n$ *and* $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n$. *Then*

$$\sum_{\mathbf{w} \in \mathbf{a}+E} (-1)^{\langle \mathbf{b}, \mathbf{w} \rangle_2} \widehat{\psi}(\mathbf{w}) \;=\; \frac{\#E}{2^n} (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle_2} \sum_{\mathbf{u} \in \mathbf{b}+E^\perp} (-1)^{\langle \mathbf{a}, \mathbf{u} \rangle_2} \psi(\mathbf{u}). \tag{2.5}$$

2.3 Walsh-Hadamard transform

For an n-variable Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, it is often convenient to apply the Fourier transform to the function $(-1)^f$ rather than f directly. The transform obtained from this is called the Walsh Transform of f.

Definition 8 (*Walsh transform*) . *The (normalised) Walsh transform of a Boolean function* f *is a map* $W_f : \mathbb{F}_2^n \to [-1, 1]$ *which is defined as follows.*

$$W_f(\boldsymbol{\alpha}) \;=\; \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus \langle \mathbf{x}, \boldsymbol{\alpha} \rangle_2} = 1 - \frac{\mathsf{wt}\left(f(\mathbf{X}) \oplus \langle \boldsymbol{\alpha}, \mathbf{X} \rangle_2\right)}{2^{n-1}}. \tag{2.6}$$

Note that the Walsh Transform measures the cross-correlations between a function and a set of linear functions. Another way to interpret the Walsh transform is through the use of Hadamard matrices.

Let H_n be the Hadamard matrix of order 2^n defined recursively as [111]

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } H_n = H_1 \otimes H_{n-1} \text{ for } n > 1$$

where $\otimes$ denotes the Kronecker product of two matrices. If we index the rows and columns of H_n by the elements of $\mathbb{F}_2^n$, we obtain $[H_n]_{(\mathbf{u},\mathbf{v})} = (-1)^{\langle \mathbf{u},\mathbf{v} \rangle_2}$. Therefore, the (normalised) Walsh transform can be written as

$$[(-1)^{f(\mathbf{0}_n)}, \ldots, (-1)^{f(\mathbf{1}_n)}]H_n = 2^n \cdot [W_f(\mathbf{0}_n), \ldots, W_f(\mathbf{1}_n)].$$

Multiplying both sides by H_n, we obtain the inverse Walsh Transform

$$(-1)^{f(\mathbf{x})} = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} W_f(\boldsymbol{\alpha})(-1)^{\langle \mathbf{x},\boldsymbol{\alpha} \rangle_2}. \tag{2.7}$$

From Parseval's theorem, we obtain the following identity.

Theorem 5 (*Parseval's identity*). *For any $f : \mathbb{F}_2^n \to \mathbb{F}_2$,*

$$\sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \left(W_f(\boldsymbol{\alpha})\right)^2 \;=\; 1. \tag{2.8}$$

So the values $\left\{\left(W_f(\boldsymbol{\alpha})\right)^2\right\}$ can be considered to be a probability distribution on $\mathbb{F}_2^n$, which assigns to $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, the probability $\left(W_f(\boldsymbol{\alpha})\right)^2$. For $k \in \{0, \ldots, n\}$, let

$$P_{\widehat{f}}(k) \;=\; \sum_{\{\mathbf{u} \in \mathbb{F}_2^n : \mathsf{wt}(\mathbf{u}) = k\}} \left(W_f(\mathbf{u})\right)^2 \tag{2.9}$$

be the probability assigned by the Fourier transform of f to the integer k.

Definition 9 (*Fourier entropy*). *The Fourier entropy $H(f)$ of f is defined to be the entropy of the probability distribution $\{W_f^2(\boldsymbol{\alpha})\}$ and is equal to*

$$H(f) = \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{F}_2^n \\ W_f^2(\boldsymbol{\alpha}) \neq 0}} W_f^2(\boldsymbol{\alpha}) \log \frac{1}{W_f^2(\boldsymbol{\alpha})}. \tag{2.10}$$

Definition 10 (*Fourier min-entropy*). *The Fourier min-entropy $H_\infty(f)$ of f is defined to*

be the min-entropy of the probability distribution $\{W_f^2(\boldsymbol{\alpha})\}$ *and is equal to*

$$H_\infty(f) = \min_{\substack{\boldsymbol{\alpha}\in\mathbb{F}_2^n \\ W_f^2(\boldsymbol{\alpha})\neq 0}} \log \frac{1}{W_f^2(\boldsymbol{\alpha})}. \tag{2.11}$$

A commonly used tool in cryptography is the auto-correlation function, which is used to calculate the correlation between a function and its shifted versions, specifically its dyadic shifts.

Definition 11 (*Auto-correlation*) . *The (normalised) auto-correlation function of* f *is a map* $C_f : \mathbb{F}_2^n \to [-1, 1]$ *defined as follows.*

$$\begin{aligned} C_f(\boldsymbol{\alpha}) &= \frac{1}{2^n}\sum_{\mathbf{x}\in\mathbb{F}_2^n}(-1)^{f(\mathbf{x})\oplus f(\mathbf{x}\oplus\boldsymbol{\alpha})} \\ &= 1 - \frac{\mathsf{wt}(f(\mathbf{X})\oplus f(\mathbf{X}\oplus\boldsymbol{\alpha}))}{2^{n-1}} = 1 - 2\Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x}\oplus\boldsymbol{\alpha})]. \end{aligned} \tag{2.12}$$

Note that $C_f(\mathbf{0}) = 1$.

For a Boolean function f, the Wiener-Khintchine formula (see Page 80 of [36]) relates the Walsh transform to the auto-correlation function.

Theorem 6 (*Wiener-Khintchine formula*) . *For any Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$,

$$\left(W_f(\boldsymbol{\alpha})\right)^2 = \widehat{C}_f(\boldsymbol{\alpha}) = \frac{1}{2^n}\sum_{\mathbf{x}\in\mathbb{F}_2^n}(-1)^{\langle\boldsymbol{\alpha},\mathbf{x}\rangle_2} C_f(\mathbf{x}). \tag{2.13}$$

Applying the inverse Fourier transform given by (2.4) to $\widehat{C}_f(\boldsymbol{\alpha})$, we obtain

$$C_f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left(W_f(\boldsymbol{\alpha})\right)^2(-1)^{\langle\boldsymbol{\alpha},\mathbf{x}\rangle}. \tag{2.14}$$

Applying (2.5) with $\psi = C_f$ and $\mathbf{a} = \mathbf{b} = \mathbf{0}_n$ and then using (2.13), we obtain the following Lemma (see Proposition 5 of [33]).

Lemma 3 *For any Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$,

$$\sum_{\mathbf{w}\in E}\left(W_f(\mathbf{w})\right)^2 = \frac{\#E}{2^n}\sum_{\mathbf{u}\in E^\perp} C_f(\mathbf{u}). \tag{2.15}$$

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of variables and suppose $\emptyset \neq \overline{T} = \{i_1, \ldots, i_{n-t}\} \subseteq [n]$, where $i_1 \leq \cdots \leq i_{n-t}$. By $\mathbf{X}_{\overline{T}}$ we denote the vector of variables $(X_{i_1}, \ldots, X_{i_{n-t}})$. Similarly for $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, $\boldsymbol{\alpha}_T$ denotes the vector $(\alpha_{i_1}, \ldots, \alpha_{i_{n-t}}) \in \mathbb{F}_2^t$. Moreover, For $T \subseteq [n]$ and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{F}_2^n$, $\boldsymbol{\alpha}_T \boldsymbol{\beta}_{\overline{T}}$ denotes the vector in $\mathbb{F}_2^n$ where the i^{th} coordinate of $\boldsymbol{\alpha}_T \boldsymbol{\beta}_{\overline{T}}$ is α_i if $i \in T$, otherwise β_i.

Suppose $f(\mathbf{X})$ is an n-variable Boolean function. For $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, by $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_T)$ we denote the Boolean function on t variables obtained by setting the variables in $\mathbf{X}_{\overline{T}}$ to the respective values in $\boldsymbol{\alpha}$. Let $f_{\boldsymbol{\alpha}}$ denote $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_T)$.

As a result of the second order Poisson summation formula (refer [36] for the general statement of this result), we obtain the following outcome.

Lemma 4 *Let $T \subseteq [n]$ with $\#T = t$. Then for any $\boldsymbol{\beta} \in \mathbb{F}_2^n$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2$, we have*

$$\sum_{\mathbf{w} \leq \chi_{\overline{T}}} \left(W_f(\boldsymbol{\beta}_T \mathbf{w}_{\overline{T}})\right)^2 \;=\; \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_\alpha}(\boldsymbol{\beta}_T)\right)^2. \tag{2.16}$$

Remark 1 *We have normalised the Walsh transform and the auto-correlation function by 2^n so that the values lie in the range $[-1, 1]$. The non-normalised versions have also been used in the literature. We note in particular that [36] uses the non-normalised versions. When we use results from [36], we normalise them appropriately.*

2.4 Influence

Definition 12 (*Influence of variable.*) *Let $f(\mathbf{X})$ be an n-variable Boolean function where $\mathbf{X} = (X_1, \ldots, X_n)$. For $i \in [n]$, the influence of X_i on f is denoted by $\mathsf{inf}_f(i)$ and is defined to be the probability (over a uniform random choice of $\mathbf{x} \in \mathbb{F}_2^n$) that $f(\mathbf{x})$ is not equal to $f(\mathbf{x} \oplus \mathbf{e}_i)$, i.e.,*

$$\mathsf{inf}_f(i) \;=\; \Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x} \oplus \mathbf{e}_i)]. \tag{2.17}$$

Lemma 5 *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$. For any $i \in [n]$,*

$$\mathsf{inf}_f(i) = \sum_{\{\mathbf{u} \in \mathbb{F}_2^n : i \in \mathsf{supp}(\mathbf{u})\}} \left(W_f(\mathbf{u})\right)^2.$$

Proof:

$$\mathsf{inf}_f(i) = \Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x}\oplus\mathbf{e}_i)] = \mathbb{E}\left[\frac{1-(-1)^{f(\mathbf{x})}\cdot(-1)^{f(\mathbf{x}\oplus\mathbf{e}_i)}}{2}\right].$$

Therefore using 2.7, we get

$$\mathsf{inf}_f(i) = \frac{1}{2}\left[1 - \sum_{\mathbf{u}\in\mathbb{F}_2^n}(W_f(\mathbf{u}))^2\cdot(-1)^{\langle u,\mathbf{e}_i\rangle_2}\right]$$

For every $\mathbf{u}$ in $\mathbb{F}_2^n$, if its support contains the index i, then $(-1)^{\langle \mathbf{u},\mathbf{e}_i\rangle_2} = -1$, otherwise $(-1)^{\langle \mathbf{u},\mathbf{e}_i\rangle_2} = 1$. Therefore, using Parseval's identity, we obtain the desired outcome. □

Definition 13 (*Total influence*) *The total influence* $\mathsf{inf}(f)$ *of* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *is defined as follows.*

$$\mathsf{inf}(f) = \sum_{i=1}^{n}\mathsf{inf}_f(i). \tag{2.18}$$

Therefore from Lemma 5 we obtain the following connection of total influence to the Walsh transform (Theorem 2.38 in [127]).

$$\mathsf{inf}(f) \;=\; \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\mathsf{wt}(\boldsymbol{\alpha})(W_f(\boldsymbol{\alpha}))^2. \tag{2.19}$$

Definition 14 (*Average sensitivity*) *The sensitivity* $\mathsf{sens}(f,\mathbf{x})$ *of a Boolean function* f *on input* $\mathbf{x} = (x_1, x_2, \ldots x_n) \in \mathbb{F}_2^n$ *is defined in the following manner.*

$$\mathsf{sens}(f,\mathbf{x}) = \#\{i \in [n] : f(\mathbf{x}) \neq f(\mathbf{x}\oplus\mathbf{e}_i)\}.$$

The average sensitivity $\mathsf{sens}(f)$ *of* f *is given by*

$$\mathsf{sens}(f) = \mathbb{E}\left(\mathsf{sens}(f,\mathbf{x})\right).$$

By linearity of expectation, average sensitivity equals the total influence.

Lemma 6 [*127*] *For any* $f : \mathbb{F}_2^n \to \mathbb{F}_2$, $\mathsf{Inf}(f) = \mathsf{sens}(f)$.

Proof:

$$\begin{aligned}\mathsf{inf}(f) &= \sum_{i=1}^{n} \Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x}\oplus\mathbf{e}_i)] \\ &= \sum_{i=1}^{n} \mathbb{E}\left(\mathbf{1}_{f(\mathbf{x})\neq f(\mathbf{x}\oplus\mathbf{e}_i)}\right) = \mathbb{E}\left(\sum_{i=1}^{n} \mathbf{1}_{f(\mathbf{x})\neq f(\mathbf{x}\oplus\mathbf{e}_i)}\right) = \mathbb{E}\left(\mathsf{sens}(f,\mathbf{x})\right) = \mathsf{sens}(f).\end{aligned}$$

□

Theorem 7 (*Poincaré inequality. [127]*) *For any* $f : \mathbf{F}_2^n \to \mathbf{F}_2$, $\mathsf{inf}(f) \geq 4\,\mathsf{Var}(f)$, *where* $\mathsf{Var}(f)$ *denotes the variance of* f.

Proof: Noting that $f^2 = f$, $\mathsf{Var}(f)$ is equal to $\mathbb{E}(f^2) - \mathbb{E}(f)^2 = \mathbb{E}(f)(1 - \mathbb{E}(f))$. Now from (2.6), $(W_f(\mathbf{0}_n))^2 = (1 - 2\mathbb{E}(f))^2$ and so $1 - (W_f(\mathbf{0}_n))^2 = 4\mathbb{E}(f)(1 - \mathbb{E}(f)) = 4\,\mathsf{Var}(f)$. Therefore, using (2.19), we have $\mathsf{inf}(f) \geq \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n\setminus\mathbf{0}_n}(W_f(\boldsymbol{\alpha}))^2 = 1 - (W_f(\mathbf{0}_n))^2 = 4\,\mathsf{Var}(f)$. □

For a subset S of the hypercube, the edge boundary of S, is the set of edges of the hypercube with one end point in S and the other endpoint outside of S. Consequently, when considering the definition of total influence, we find that total influence is n times the (fractional) size of the edge boundary of $\mathsf{supp}(f)$. Moreover, since $\mathsf{Var}(f) = \frac{\#\mathsf{supp}(f)(1-\#\mathsf{supp}(f))}{2^{2n}}$, is linked to the size of the set $\mathsf{supp}(f)$. Therefore, the Poincaré Inequality can be viewed as an edge-isoperimetric inequality.

2.5 Noise stability

In a 2-candidate election, the voting rule f maps each voter's vote $\mathbf{x} = (x_1, \ldots, x_n)$ from the set $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Each voter's vote may be recorded incorrectly with a probability of $1 - \rho$, where ρ is the probability of correct recording. The recorded votes, $\mathbf{y} = (y_1, \ldots, y_n)$, may differ from the original votes $\mathbf{x}$. The noise stability of f refers to the probability that the outcome of the election, given by $f(\mathbf{x})$, will remain the same as $f(\mathbf{y})$ despite the potential misrecording of the votes.

Definition 15 (ρ-*correlated pair*) *For* $\mathbf{x} \in \mathbf{F}_2^n$ *and* $\rho \in [0, 1]$, *define a distribution* $N_\rho(\mathbf{x})$ *over* $\mathbb{F}_2^n$ *in the following manner:* $\mathbf{y} = (y_1, \ldots, y_n) \sim N_\rho(\mathbf{x})$ *if for* $i = 1, \ldots, n$,

$$y_i = \begin{cases} x_i, & \textit{with probability } \rho \\ 0/1, & \textit{with probability } \frac{1-\rho}{2} \textit{ each.} \end{cases} \tag{2.20}$$

The definition of $N_\rho(\mathbf{x})$ can be extended to all values of ρ in the range $[-1, 1]$, as follows [127].

$$y_i = \begin{cases} x_i, & \textit{with probability } \frac{1}{2} + \frac{\rho}{2} \\ 1 \oplus x_i, & \textit{with probability } \frac{1}{2} - \frac{\rho}{2}. \end{cases} \tag{2.21}$$

We say that $(\mathbf{x}, \mathbf{y})$ is a ρ-correlated pair of random strings.

Remark 2 *An equivalent way of defining a ρ-correlated pair of random strings is by stating that for each $i \in [n]$, the two random bits (x_i, y_i) are independent with expected values $\mathbb{E}(x_i) = \mathbb{E}(y_i) = \frac{1}{2}$ and $\mathbb{E}((-1)^{x_i} \cdot (-1)^{y_i}) = \rho$.*

Definition 16 (*Noise operator*) *For $\psi : \mathbb{F}_2^n \to \mathbb{R}$, the noise operator with parameter $\rho \in [-1, 1]$ on ψ, denoted by $T_\rho : \mathbb{F}_2^n \to \mathbb{R}$ is defined as follows:*

$$T_\rho \psi(\mathbf{x}) = \mathop{\mathbb{E}}_{\mathbf{y} \sim N_\rho(\mathbf{x})} \left(\psi(\mathbf{y})\right). \tag{2.22}$$

Lemma 7 *[127] For $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_2^n$, $\widehat{T_\rho \psi}(\boldsymbol{\alpha}) = \rho^{\mathsf{wt}(\boldsymbol{\alpha})} \widehat{\psi}(\boldsymbol{\alpha})$.*

Proof: For any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, if we apply T_ρ on $\mathcal{C}_{\boldsymbol{\alpha}}$, then we obtain the following:

$$T_\rho \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) = \mathop{\mathbb{E}}_{\mathbf{y} \sim N_\rho(\mathbf{x})} \left((-1)^{\langle \mathbf{y}, \boldsymbol{\alpha} \rangle_2}\right) = \prod_{i:\alpha_i = 1} \mathop{\mathbb{E}}_{\mathbf{y} \sim N_\rho(\mathbf{x})} \left((-1)^{y_i}\right) = \prod_{i:\alpha_i = 1} \left(\rho \cdot (-1)^{x_i}\right) = \rho^{\mathsf{wt}(\boldsymbol{\alpha})} \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}).$$

Therefore as a result of (2.4) and the fact that T_ρ is a linear operator, the following holds,

$$T_\rho \psi\left(\mathbf{x}\right) = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \widehat{\psi}(\boldsymbol{\alpha}) T_\rho \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \rho^{\mathsf{wt}(\boldsymbol{\alpha})} \widehat{\psi}(\boldsymbol{\alpha}) \mathcal{C}_{\boldsymbol{\alpha}}(\mathbf{x})$$

The above equation gives us the Fourier representation of $T_\rho \psi$. Therefore, the Fourier coefficient of $T_\rho \psi$ at $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ i.e. $\widehat{T_\rho \psi}(\boldsymbol{\alpha})$ is equal to $\rho^{\mathsf{wt}(\boldsymbol{\alpha})} \widehat{\psi}(\boldsymbol{\alpha})$. □

Having established these definitions, the concept of noise stability can now be defined.

Definition 17 (*Noise stability*) *For $\rho \in [-1,1]$, the noise stability of a function $\psi : \mathbb{F}_2^n \to \mathbb{R}$, denoted by $\mathsf{Stab}_\rho(\psi)$, is defined as follows.*

$$\mathsf{Stab}_\rho(\psi) = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{F}_2^n,\ \mathbf{y}\sim N_\rho(\mathbf{x})} [\psi(\mathbf{x})\psi(\mathbf{y})]\,. \tag{2.23}$$

Taking into consideration Equations (2.23) and (2.22), we can conclude that the noise stability of ψ can be expressed as $\mathsf{Stab}_\rho(\psi) = \langle \psi, T_\rho \psi \rangle$. Based on this, Plancherel's Theorem and Lemma 7 yield the following conclusion.

Lemma 8 $\mathsf{Stab}_\rho(\psi) = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \rho^{\mathsf{wt}(\boldsymbol{\alpha})} \cdot \left(\widehat{\psi}(\boldsymbol{\alpha})\right)^2$.

In the voting scenario described, when ρ is near 1, indicating a low level of "noise", it is often appropriate to investigate the chance that flipping a portion of the votes will result in a change of the election outcome, thereby evaluating the noise sensitivity of the voting rule.

Definition 18 (*Noise sensitivity*) *For $\delta \in [0,1]$, the noise sensitivity of an n-variable Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, denoted by $\mathsf{NS}_\delta(f)$, is defined as follows.*

$$\mathsf{NS}_\delta(f) = \Pr_{\mathbf{x}\sim\mathbb{F}_2^n,\ \mathbf{y}\sim N_{(1-2\delta)}(\mathbf{x})} [f(\mathbf{x} \neq f(\mathbf{y}))]\,.$$

It is easy to show that (see [127] for proof),

Lemma 9 *For any $\delta \in [0,1]$ and $f : \mathbb{F}_2^n \to \mathbb{F}_2$, $\mathsf{NS}_\delta(f) = \frac{1}{2} - \frac{1}{2}\mathsf{Stab}_{(1-2\delta)}\left((-1)^f\right)$.*

2.6 Some well-known inequalities

Minkowski inequality. For any functions $\psi_1, \psi_2 : \mathbb{F}_2^n \to \mathbb{R}$ and some real number $p > 1$, $\|\psi_1 + \psi_2\|_p \leq \|\psi_1\|_p + \|\psi_2\|_p$.

Hoeffding's inequality. Let $X_1, \ldots, X_n$ be independent bounded random variables with $X_i \in [a,b]$ for all i, where $-\infty < a \leq b < \infty$. Then,

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}(X_i))\right| \geq \epsilon\right] \leq \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right).$$

Chapter 3

Fourier Analysis on Boolean Hypercube

The Fourier transform is a fundamental concept in mathematics and computer science that has numerous applications across a variety of fields. In the study of Boolean functions, the Fourier/Walsh transform on Boolean hypercube has proven to be a critical tool and has gained significant importance in the fields of cryptography and theoretical computer science. Although this chapter covers some of its applications, they are not exhaustive, and a comprehensive book would be necessary for further study. Two highly recommended books for studying Boolean functions are [127] and [36]. The first book focuses on Boolean functions in the context of theoretical computer science, while the second book focuses on Boolean functions in relation to cryptography and coding theory.

3.1 Property testing

According to Goldreich [70], the concept of property testing in theoretical computer science was inspired by the work [25] of Blum, Luby, and Rubinfeld. Blum et. al. in [25] showed that for any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, it is possible to differentiate between the case that f is linear and the case that it needs to be modified on at least a fraction of $\epsilon > 0$ points to become linear, through a few random inquiries. Rubinfeld and Sudan in [146] introduced the first general definition of property testing, as follows: Let $\mathcal{P} = \cup_{n\in\mathbb{N}}\mathcal{P}_n$, where $\mathcal{P}_n$ is a collection of n-variable Boolean functions. A tester for a "property" $\mathcal{P}$ is a randomized algorithm $\mathcal{T}$, which has query access to any unknown $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and satisfies the following two conditions:

1. If $f \in \mathcal{P}$, then $\mathcal{T}$ accepts f with probability at least $\frac{2}{3}$.
2. Given $\epsilon > 0$, if f is ϵ-far from $\mathcal{P}$ that is $\min_{g\in\mathcal{P}} \Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq g(\mathbf{x})] \geq \epsilon$, then the tester rejects with probability at least $\frac{2}{3}$.

If the tester accepts every function in $\mathcal{P}$ with probability 1, we refer to it as having one-sided error. This convention was also used by Goldreich, Goldwasser, and Ron in their

work [71], where they demonstrated that many graph properties are testable. They regarded graphs as Boolean functions applied to pairs of vertices, with the function's value indicating the presence of an edge. Under the uniform distribution, a testing algorithm for a graph property $\mathcal{P}$ conducts queries in the form of "does an edge exist between vertices u and v" within an unknown graph G. Consequently, the distance between two N-vertex graphs is quantified as the fraction (out of N^2) of vertex pairs that are adjacent in one graph but not in the other. As earlier testing algorithm for property $\mathcal{P}$ is tasked with determining whether G possesses property $\mathcal{P}$ or is ϵ-far from any graph exhibiting property $\mathcal{P}$, with some permissible error probability. Since then, numerous studies have been conducted to classify testable properties. In this section, we discuss some results in the field of property testing that are based on Fourier analysis.

Linearity testing. Any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is said to be linear if $f(\mathbf{x} \oplus \mathbf{y}) = f(\mathbf{x}) \oplus f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$. It is trivial to show that every linear function is of the form $l_{\boldsymbol{\alpha}}(\mathbf{x}) = \langle \boldsymbol{\alpha}, \mathbf{x} \rangle_2$, where $\boldsymbol{\alpha} \in \mathbb{F}_2^n$.

BLR test. Given query access to $f : \mathbb{F}_2^n \to \mathbb{F}_2$.

1. Choose two random points $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$.
2. Accept if $f(\mathbf{x}) \oplus f(\mathbf{y}) = f(\mathbf{x} \oplus \mathbf{y})$, otherwise reject.

The original analysis by Blum et. al. [25] was combinatorial, but there is also a Fourier analytic approach presented by Bellare et. al. [11]. The bulk of this analysis lies in proving that if f is ϵ-far from every linear function, then f is accepted with some negligible probability. We briefly discuss the approach of [11] here.

Bellare et. al. in [11], showed that the probability that f is not rejected by the BLR test is upper bounded by $\frac{1}{2} + \frac{1}{2} \max_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} W_f(\alpha)$. Note that, if f is ϵ-far from every linear function then for every $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, $\Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq l_{\boldsymbol{\alpha}}(\mathbf{x})] \geq \epsilon$ that is $\frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \mathsf{wt}(f(\mathbf{x}) \oplus \langle \boldsymbol{\alpha}, \mathbf{x} \rangle_2) \geq \epsilon$. Now from equation 2.6, it means that $W_f(\alpha) \leq 1 - 2\epsilon$ for any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. Therefore from Bellare et. al.'s result, if f is ϵ-far from every linear function then $\Pr_{\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n}[f(\mathbf{x}) \oplus f(\mathbf{y}) = f(\mathbf{x} \oplus \mathbf{y})] \leq 1 - \epsilon$.

This is a one-sided error test, which always accepts f if f is linear. The success rate of the test can be increased by repeating it multiple times. Specifically, by running the test $\frac{2}{\epsilon}$ times independently, f is accepted if it is accepted in all $\frac{2}{\epsilon}$ trials, otherwise it is rejected. The

total number of queries required is $\frac{6}{\epsilon}$. For f that is ϵ-far from being linear, the acceptance rate is at most $(1-\epsilon)^{\frac{2}{\epsilon}} \leq \frac{1}{3}$, which satisfies the conditions of the Rubinfeld and Sudan's definition of property testing.

Dictator testing. Now, we examine a simpler property referred to as being a dictator. Specifically, we examine the following properties defined as follows:

$$\mathcal{D} = \{f : \mathbb{F}_2^n \to \mathbb{F}_2 : f(x_1, \ldots, x_n) = x_i \text{ for some } i \in [n]\}.$$

The complemented-dictatorship property, denoted by $\mathcal{D}^c$, has been defined as follows:

$$\mathcal{D}^c = \{f : \mathbb{F}_2^n \to \mathbb{F}_2 : f(x_1, \ldots, x_n) = 1 \oplus x_i \text{ for some } i \in [n]\}.$$

The existence of a tester for dictatorship cannot be automatically inferred from the fact that $\mathcal{D}$ is a subset of linear functions and that there is a BLR tester for linearity. This is because, even if a linear function $l_{\boldsymbol{\alpha}}$ is not a dictator (i.e. $\mathsf{wt}(\boldsymbol{\alpha}) \neq 1$), the probability that $l_{\boldsymbol{\alpha}}(\mathbf{x}) \neq g(\mathbf{x})$ for any $g \in \mathcal{D}$ is equal to $\frac{1}{2}$, but the BLR test will still accept it with a probability of 1. This is different from the principle in learning theory where a learning algorithm for a class can be applied to any of its subclasses automatically.

However, the BLR test can be utilized as a preliminary step in testing for dictatorship. This reduces the task of determining whether an unknown linear function is a dictator. The first testers for dictatorship, presented in [10] and [132], used the following method: after confirming linearity, the testers randomly choose $\mathbf{x}$ and $\mathbf{y}$ from $\{0,1\}^n$, calculate $\mathbf{z} = \mathbf{x} \wedge \mathbf{y}$, and verify if $f(\mathbf{z}) = f(\mathbf{x}) \wedge f(\mathbf{y})$. The only functions that pass this "AND test" with probability 1 are dictators (and constant functions). To handle the constant function case, some changes to the testing algorithms and a more sophisticated analysis of the test are required. An alternative approach of dictator testing is to use the Arrow's famous result [4], which characterizes the property $\mathcal{D} \cup \mathcal{D}^c$. This characterisation leads to a simpler test known as the "Not-All-Equal test" [127].

Not-All-Equal test. Given query access to $f : \mathbb{F}_2^n \to \mathbb{F}_2$.

1. Choose $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_2^n$ such that (x_i, y_i, z_i) are drawn independently and uniformly at random from $\{(0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0)\}$.

2. Accept if $(f(\mathbf{x}), f(\mathbf{y}), f(\mathbf{z})) \notin \{(0,0,0),(1,1,1)\}$, otherwise reject.

Under the "impartial culture assumption" for 3-candidate elections, where each voter independently chooses one of the 6 possible rankings uniformly at random, Arrow's Theorem was later proved in 2002 using Fourier analysis by Kalai in [93]. In [93] Kalai showed that the above test accepts f with probability $\frac{3}{4} - \frac{3}{4}\sum_{k=0}^{n}(-\frac{1}{3})^k P_{\widehat{f}}(k)$. Therefore, if the "Not-All-Equal test" accepts f with probability $1-\epsilon$, then from Kalai's result, it is possible to show that $P_{\widehat{f}}(1) \geq 1 - \frac{9}{2}\epsilon$ [127]. This means that f is $O(\epsilon)$ close to either being a dictator or a complemented-dictator, as determined by the Friedgut-Kalai-Naor Theorem [67].

However, there are limitations to this testing. It only gives a local tester for the property of being a dictator or complemented-dictator, and the conclusion about f being close to such a property relies on the non-trivial Friedgut-Kalai-Naor Theorem. To address these issues, the BLR Test can be added with the "Not-All-Equal test", as discussed in [127].

k-Junta testing. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is considered to be a k-junta if it has at most k "influential" variables [23]. Alternatively, f is said to be an k-junta if there is a subset $S \subseteq [n]$ with $\#S \leq k$ such that f is degenerate on the variables indexed by $\overline{S}$. Parnas, Ron, and Samorodnitsky [132] were the first to demonstrate a testing algorithm for 1-juntas, that is for dictators and completed dictators, with only $O(1/\epsilon)$ queries, by generalizing a result from Bellare, Goldreich, and Sudan [10] on testing long codes. Fischer et al. then proposed algorithms with 1-sided error for testing k-juntas, using $O(k^4 \ln(k+1)/\epsilon)$ queries, and a better algorithm with $O(k^2 \ln^2(k+1)/\epsilon)$ query complexity and a 2-sided error. The best known junta-testing algorithm with 1-sided error to date, by Blais [23], requires $O(k/\epsilon + k \log k)$ queries.

The first significant lower bound on the query complexity of junta testing was established by Fischer et al. [62], who demonstrated that $\omega(\sqrt{k})$ queries are necessary to test k-juntas. This lower bound was later improved to $\omega(k)$ by Chockler and Gutfreund [44].

Therefore, based on the lower bound established by Chockler and Gutfreund, the algorithm proposed by Blais in [23] is nearly optimal in terms of query complexity, with only a logarithmic factor difference. A brief overview of the algorithm is provided below.

Blais k-junta testing. The Blais junta testing algorithm is based on a key observation made by Blum, Hellerstein, and Littlestone in their work [24]. If there exists two inputs $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{F}_2^n$ such that $f(\mathbf{x}) \neq f(\mathbf{y})$, then the set of variables (say S) where $\mathbf{x}$ and $\mathbf{y}$

differ must contain a influential variable for f. Is it possible to utilize a binary search on the 'hybrid inputs', denoted as $\mathbf{z}$, created from $\mathbf{x}$ and $\mathbf{y}$ in the following manner: $z_i = x_i$ if $i \in T \subseteq [S]$, $z_i = y_i$ if $i \in S \setminus T$, and $z_i = x_i = y_i$ otherwise. Then the influential variable can be identified with $O(\log |S|)$ queries.

Blais's approach involves maintaining a set S of variables that may or may not be influential to the function. Given a randomly generated pair of inputs $(\mathbf{x}, \mathbf{y})$ that satisfies the above criteria over S, instead of identifying the influential variable, Blais aimed to identify the part containing the influential variable in a random partition of the set S into poly(k) parts. This can also be done through a binary search in $O(\log k)$ queries. The algorithm then removes all variables in that part from S. If the algorithm identifies $k+1$ different parts with influential variables, the function is rejected; otherwise, it is accepted.

It is easy to see that if the input function is a k-junta, it will contain at most k parts with influential variables, and the Blais algorithm will accept it. Thus, the algorithm has one-sided error. The soundness of the algorithm is established through the following argument: if a function that is ϵ-far from being a k-junta, it will be $\frac{\epsilon}{2}$-far from being a "k-part junta" (functions whose influential variables are contained within no more than k parts with respect to a random partition of variables) with high probability. To prove the aforementioned lemma, it is necessary to first understand the concept of an influential set of variables. Blais employs the concept of "influence over a set of variables S" (say $I_f(S)$), introduced by Fischer et al. in [62]. Using this notion of influence, Blais established the following characterization of non-juntas, which forms the core of his main lemma:

Lemma 10 *[23] If f is ϵ-far from being an k-junta, then for any set $S \subseteq [n]$ with $\#S \leq k$, $I_f(\overline{S}) \geq \epsilon$.*

However, a more straightforward proof of the above characterisation is presented in Chapter 5.

3.2 The Bonami-Beckner hypercontractive inequality and some of its applications

This section discusses a significant technique utilized in harmonic analysis, known as the hypercontractivity of the noise operator. This approach was developed by Bonami [26] and Beckner [9].

In Chapter 2 (see Relation 2.22), we discussed the noise operator. As shown in Lemma 7, the noise operator T_ρ dampens the higher-degree Fourier coefficients, and this reduction effect grows exponentially with the degree of the coefficients. Essentially, the noise operator has a "smoothing" impact on the function ψ. When $\rho = 1$, the result of $T_\rho\psi$ is identical to ψ, but as ρ decreases, the function $T_\rho\psi$ approaches the constant $\mathbb{E}(\psi)$. In fact, when $\rho = 0$, $T_\rho\psi$ equals $\mathbb{E}(\psi)$. For $\rho < 1$, only the constant functions ψ satisfy $T_\rho\psi = \psi$.

The operator $T_\rho\psi$ computes an average of functions that possess the same p-norm as ψ. Therefore, according to the Minkowski inequality, T_ρ is a contraction, indicating that for all $p \leq 1$, $\|T_\rho\psi\|_p \leq \|\psi\|_p$. It is worth noting that the p-norm is monotone non-decreasing in p, meaning that $\|\psi\|_p \leq \|\psi\|_q$ for $1 \leq p \leq q \leq \infty$. The Bonami-Beckner Hypercontractive inequality [26, 9] shows that if we sufficiently smooth the function f using T_ρ, we can reverse the direction of the second inequality mentioned. In other words, T_ρ is not only contractive but also hypercontractive. Specifically, for $1 \leq p \leq q$ and $0 \leq \rho \leq \sqrt{\frac{p-1}{q-1}}$, the inequality $\|T_\rho\psi\|_q \leq \|\psi\|_p$ holds true.

The Bonami-Beckner inequality has several significant applications in the analysis of Boolean functions. The following are a few examples.

Structures of functions with small total influence. Recall that a Boolean function on $\mathbb{F}_2^n$ that depends on at most k of its variables is called a k-junta. If f is a k-junta, then its influence is at most k. This is because each variable that f does not depend on has an influence of 0, and every other variable has an influence of at most 1. Friedgut's junta theorem [65] partially reverses this observation, as it asserts that a Boolean function with a small total influence can be accurately approximated by a k-junta, where k is also small. In the proof of this theorem, the Walsh spectrum of f is separated into high-weighted characters and low-weighted characters. Although the high-weighted characters is easy to handle (similar to lemma 11), the Bonami-Beckner Hypercontractive inequality is required to bound the low-weighted characters.

Lower bound of $\max_i \mathsf{Inf}_f(i)$ for balanced functions. The concept of influence of a variable on a Boolean function was first introduced by Ben-Or and Linial [12] in the context of a collective coin flipping scheme. In this scheme, each player picks a random bit x_i and the value of the coin flip is determined by the function $f(\mathbf{x})$. To maintain security in the collective coin flipping scheme, it is crucial to prevent small collusions of dishonest players, who can observe the bits of honest players, from significantly affecting the output value of

the function f.

There are two extreme cases of functions with regard to influence: the constant function (where $\mathsf{Inf}_f(i) = 0$ for all i) and the parity function (where $\mathsf{Inf}_f(i) = 1$ for all i). The dictator function $f(\mathbf{x}) = x_i$ is another example where only the i^{th} variable has influence 1 while all other influences are 0. Another significant example is the maj_n, where each variable has an influence of $\theta(\frac{1}{\sqrt{n}})$.

The Poincaré inequality (see Relation 7) for balanced functions implies that the influence of a variable on a Boolean function is at least 1. This leads to the natural question of whether it is possible to find balanced functions where the variable influences are much smaller, ideally on the order of $O(\frac{1}{n})$. Let $\mathcal{f} : \{0,1\}^n \to \{0,1\}$ be defined as follows. $f(\mathbf{X}) = \bigvee_{j=1}^{m} \bigwedge_{i=(j-1)k+1}^{(j+1)k} X_i$, where $m = \frac{n}{k}$. $\mathcal{f}$ is known as "tribes DNF formulas". Since, $\mathsf{Pr}_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) = 1] = 1-(1-2^{-k})^m$, so $\mathcal{f}$ would be balanced or nearly balanced if $m \approx (\ln 2)2^k$, k is some integer. Hence, $n \approx (\ln 2)k2^k$, therefore, $k \approx \log n - \log\log n$ and $m \approx \frac{n}{\log n}$. Considering the above parameter Ben-Or and Linial [12] showed that each variable in $\mathcal{f}$ has influence $\theta(\frac{\log n}{n})$. Kahn, Kalai, and Linial [92] later proved that this is nearly optimal, showing that for any balanced or nearly balanced Boolean function, at least one variable has influence $\omega(\frac{\log n}{n})$. The proof is similar to the one used in Friedgut's junta theorem [65]. This observation implies that a set of only $O(\frac{n}{\log n})$ variables can determine the function value for almost all settings of the other variables with high probability [2]. Hence, these functions are not suitable for collective coin flipping protocols that are secure against a constant fraction of dishonest players, as a small group of colluding players can easily manipulate the result [52]. Talagrand [165] provided a slightly stronger result, and Bourgain et. al. [28] extended the KKL result to n-variable Boolean functions with real-valued inputs, where each real-valued variable has uniform measure.

3.3 Learning theory

"Probably Approximately Correct" (PAC) learning, introduced by Valiant in the 1980s [169], provides a structured framework for computational learning. In the PAC learning model, which operates under the uniform distribution on $\mathbb{F}_2^n$, the central focus is on a concept class denoted as $\mathcal{C}$. This concept class serves as a collection of functions, map inputs from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, possessing specific properties. Learning a concept class means, for any Boolean function f within $\mathcal{C}$, utilizing it as an oracle, trying to seek a hypothesis (any Boolean function but may not necessarily belong to $\mathcal{C}$) that effectively predicts $f(\mathbf{y})$ for future inputs $\mathbf{y}$. A learning

algorithm is considered PAC-learnable if it can generate a hypothesis, denoted as h, that is likely to be mostly correct with a high probability, given limited access to a sufficiently large set of samples. To elaborate on this concept, let us imagine a scenario where we aim to learn from a concept class $\mathcal{C}$. In this scenario, we have access to a limited set of samples in the form of $(\mathbf{x}, f(\mathbf{x}))$ pairs, where f is an unknown function belonging to $\mathcal{C}$. We introduce a target error parameter ϵ, which falls within the range of 0 to $\frac{1}{2}$, and a confidence parameter δ that is greater than 0. For any $f \in \mathcal{C}$, we define a learning algorithm as PAC-learnable if it can find a sample of size m such that, with a probability of at least $1-\delta$, the hypothesis h it generates satisfies the condition $\Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq h(\mathbf{x})] \leq \epsilon$. This means that, with high confidence, the hypothesis function h is not "far" from the target function f. In PAC learning, there are two primary access models. The first one involves random examples, where the learning algorithm can draw pairs $(\mathbf{x}, f(\mathbf{x}))$ with $\mathbf{x} \in \mathbb{F}_2^n$ chosen uniformly at random. The second access model is through queries, allowing the algorithm to request the value $f(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{F}_2^n$ of its choice. These two access models differ in their strength, with the query model being more powerful than random examples.

Efficient running time is the primary requirement of a learning algorithm. While it is possible to learn any function f with zero error in a time complexity of $O(n \cdot 2^n)$ [127], this approach is not efficient. When the concept class $\mathcal{C}$ includes highly complex functions, which are those with well-distributed Walsh coefficients, exponential running time is necessary. However, if $\mathcal{C}$ comprises relatively "simple" functions, it may be possible to achieve more efficient learning. To understand what is meant by "simple" functions, it is necessary to introduce a definition.

Definition 19 (ϵ-*concentration on* $\mathcal{F}$) *[127] Let* $\mathcal{F} \subseteq \mathbb{F}_2^n$. *Given an* n*-variable Boolean function* f*, we say that the Walsh spectrum of* f *is* ϵ*-concentrated on* $\mathcal{F}$ *if* $\sum_{\substack{\boldsymbol{\alpha}\in\mathbb{F}_2^n \\ \boldsymbol{\alpha}\notin\mathcal{F}}} W_f(\boldsymbol{\alpha})^2 \leq \epsilon$.

If the size of $\mathcal{F}$ is small, a Boolean function can be considered as "simple". Suppose that a learning algorithm A has random example access to the target function $f : \mathbb{F}_2^n \to \mathbb{F}_2$. If A can somehow identify a collection $\mathcal{F} \subseteq \mathbb{F}_2^n$ on which f's Walsh spectrum is $\frac{\epsilon}{2}$-concentrated, then a common way of attempting to learn f is to estimate all of f's Walsh coefficients in $\mathcal{F}$. As the Walsh coefficient at any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ is just an expectation under the uniform distribution on $\mathbb{F}_2^n$ (see 2.6), it can be approximated from uniformly drawn examples $(\mathbf{x}^{(1)}, f(\mathbf{x}^{(1)})), \ldots, (\mathbf{x}^{(m)}, f(\mathbf{x}^{(m)}))$. For any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, the empirical average $\frac{1}{m}\sum_{i=1}^{m}(-1)^{f(\mathbf{x}^{(i)})\oplus\langle\mathbf{x}^{(i)},\boldsymbol{\alpha}\rangle_2}$ will converge to the correct value $W_f(\boldsymbol{\alpha})$ as m grows. Using Hoeffding's inequality, it can be shown that obtaining an estimate $W'_f(\boldsymbol{\alpha})$ for $W_f(\boldsymbol{\alpha})$ within an error of $\pm\epsilon$ can be achieved with probability at least $1-\delta$ using

$O(\frac{1}{\epsilon^2}\log(\frac{1}{\delta}))$ examples. Finally, using the estimated coefficients, A forms the real-valued function $\psi(\mathbf{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{F}} W'_f(\boldsymbol{\alpha})(-1)^{\langle \mathbf{X},\boldsymbol{\alpha}\rangle_2}$ and outputs hypothesis $h(\mathbf{X}) = \frac{1-\mathsf{sign}(\psi(\mathbf{X}))}{2}$, where $\mathsf{sign}(z) = 1$ if $z \geq 0$, and -1 if $z < 0$. Given the assumption that the Walsh spectrum of the target function f is $\frac{\epsilon}{2}$-concentrated on the set $\mathcal{F}$, the method described above results in a reliable approximation of f. Specifically, the hypothesis h obtained from the algorithm is ϵ-close to the target concept f or $\Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq h(\mathbf{x})] \leq \epsilon$ [127].

The "Low-Degree algorithm". There are concept classes for which we can choose $\mathcal{F}$ without any algorithmic searching, by simply taking all vectors of small weights. This approach is effective when all functions in $\mathcal{C}$ exhibit spectral concentration at small weights. The "Low-Degree Algorithm", developed by Linial, Mansour, and Nisan in their pioneering work on the Fourier approach to computational learning [110], can be used to learn such concept classes. However, before delving into this algorithm, we need to first define the weighted version of spectral concentration.

Definition 20 (ϵ*-concentration weights up to* k)*.* *Given an* n*-variable Boolean function* f*, we say that the Walsh spectrum of* f *is* ϵ*-concentrated on coefficients of weights up to* k *if* $\sum_{i\geq k} P_{\hat{f}}(i) \leq \epsilon$*.*

The "Low-Degree Algorithm" is based on the idea of reducing the learning problem of a concept class $\mathcal{C}$ from random examples to the analytical task of showing small weight spectral concentration for the functions in $\mathcal{C}$. Specifically, if we can prove that for some $k \geq 1$, every function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ in $\mathcal{C}$ is $\frac{\epsilon}{2}$-concentrated up to weight k, then we can set $\mathcal{F}$ to be the set of all vectors of weight at most k, which has size at most $O(n^k)$. We can then approximate $\mathcal{C}$ well by a real polynomial of low degree using the Walsh coefficients estimation approach described earlier.

For instance, in [110], Linial, Mansour, and Nisan showed that if a Boolean function f is computable by a bounded depth circuit [83] of depth d and size M, then f can be well approximated by a "polynomial threshold function (PTF)" [30], of degree at most $O(\log(M/\epsilon)^d)$. This approach thus provides a systematic way to learn concept classes with small-weight spectral concentration without the need for algorithmic searching.

One way to demonstrate such a concentration result is by proving combinatorially that a function has small total influence. This can be done using the following lemma.

Lemma 11 *For any* $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *and* $\epsilon > 0$*, the Walsh spectrum of* f *is* ϵ*-concentrated on coefficients of weights up to* $\frac{\mathsf{inf}(f)}{\epsilon}$*.*

Using Lemma 11, we can quickly obtain some learning-theoretic results. For example, consider the concept class $\mathcal{C} = \{f : \mathbb{F}_2^n \to \mathbb{F}_2 \mid f \text{ is monotone}\}$. We can learn $\mathcal{C}$ from random examples with error ϵ in time $n^{O\left(\frac{\sqrt{n}}{\epsilon}\right)}$, since $\mathsf{inf}(f) \leq O(\sqrt{n})$ [177]. It might be concerning that a running time such as $n^{O(\sqrt{n})}$ does not seem very efficient. However, it is much better than the trivial running time of $O(n \cdot 2^n)$.

Another way to establish low-degree spectral concentration is by analyzing noise stability or sensitivity. For any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and $\delta \in [0, \frac{1}{2}]$, the Walsh spectrum of f is $3\mathsf{NS}_\delta(f)$-concentrated on degrees up to $\frac{1}{\delta}$. The noise sensitivity approach was introduced by Klivans, O'Donnell, and Servedio in [103] as an alternative way to the Low-Degree Algorithm based on concentration results from Lemma 11.

Using noise sensitivity can be more effective in some scenarios. For example, it tells us that for $\delta > 0$ sufficiently small and n sufficiently large (as a function of δ), the Walsh spectrum of the n-variable "majority" function is $3\sqrt{\delta}$-concentrated on degrees up to $\frac{1}{\delta}$ since $\mathsf{NS}_\delta(\mathsf{maj}_n) \leq \sqrt{\delta}$ [127]. This is equivalent to saying that it is ϵ-concentrated on degrees up to $O(\frac{1}{\epsilon^2})$. In contrast, Lemma 11 gives us that the "majority" function is concentrated on weights up to $O(\frac{\sqrt{n}}{\epsilon})$ since it is a monotone function.

Goldreich–Levin algorithm. The work by Goldreich and Levin was published in 1989 [72]. Apart from its relevance in cryptography and learning, it also has significant implications in coding theory and complexity as a local list-decoding algorithm for the Hadamard code. Despite being initially developed for cryptography to build a "pseudorandom generator" from a "one-way permutation", the Goldreich-Levin algorithm found its way into the field of learning theory due to Kushilevitz and Mansour for learning decision trees [109]. The approach to learning based on estimating Walsh coefficients reduces the problem of learning an unknown target function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ to identifying a collection $\mathcal{F} \in \mathbb{F}_2^n$ that is $\frac{\epsilon}{2}$-concentrated. Similar to estimating the Walsh coefficients, we can also almost accurately estimate the Fourier probability of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ at any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, represented as $\left(W_f(\boldsymbol{\alpha})\right)^2$. By determining whether this value is large or small, we can decide to keep or discard it, respectively. This iterative process allows us to identify the desired collection $\mathcal{F}$ of candidates. However, due to the large number of candidates (2^n), identifying the desired collection $\mathcal{F}$ can be difficult. To overcome this challenge, the Goldreich-Levin algorithm employs a divide-and-conquer strategy that estimates the partial sum of the Fourier probabilities of f. A brief overview of this algorithmic approach will be provided here, but for a more detailed analysis, refer to [127].

Consider a "bucketing system" defined as follows: for $0 \leq t \leq n$ and $S \subseteq T \subseteq [n]$ with $\#T = t$, the bucket $\mathcal{B}_{t,S}$ consists of all sets that are of the form $S \cup R$, where $R \subseteq \{t+1, t+2, \ldots, n\}$. Note that the number of elements in each bucket $\mathcal{B}_{t,S}$ is $\frac{1}{2^{n-t}}$. The initial bucket is $\mathcal{B}_{0,\phi}$, and the algorithm always splits a bucket $\mathcal{B}_{t,S}$ into two buckets: $\mathcal{B}_{t+1,S}$ and $\mathcal{B}_{t+1,S\cup\{t+1\}}$. The final singleton buckets are of the form $\mathcal{B}_{n,S} = S$. The partial sum of Fourier probabilities at $\mathcal{B}_{t,S}$ is defined as $\sum_{U\in\mathcal{B}_{t,S}} \left(W_f\left(\chi_U\right)\right)^2$, which is precisely equal to $\sum_{\mathbf{w}\leq\chi_{\overline{T}}} \left(W_f(\boldsymbol{\beta}_T \mathbf{w}_{\overline{T}})\right)^2$, where $\boldsymbol{\beta} = \chi_S$ and by the second order Poisson summation formula (2.16), this is equal to $\frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^{n-t}} \left(W_{f_\alpha}(\boldsymbol{\beta}_T)\right)^2$. It is worth noting that if we have an oracle access to $f_{\boldsymbol{\alpha}}$ for any $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, then we can estimate $\left(W_{f_\alpha}(\boldsymbol{\beta}_T)\right)^2$ using random examples, as we did earlier. However, since we only have access to the oracle for f, we require query access, where the values of the coordinates in a query $\mathbf{x} \in \mathbb{F}_2^n$ corresponding to $\overline{T}$ are fixed to some $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$.

The above algorithm is often described as the Kushilevitz-Mansour algorithm. Unlike the Low-Degree Algorithm, which requires that the Walsh spectrum of f is concentrated on low-weighted characters, the Kushilevitz-Mansour Algorithm works as long as the spectrum is concentrated on some small collection of sets. However, a drawback of the Kushilevitz-Mansour Algorithm is that it requires query access to f, instead of just random examples.

Several other computational learning results have also utilized these ideas, and some examples of these results include [114, 88, 104, 31, 123, 129].

3.4 Threshold phenomena

A "sharp threshold" in graph theory refers to the property of a certain graph property that exhibits a sudden transition from almost no instances of the property to almost all instances of the property as the size of the graph increases. Specifically, for the random graph model $G(n,p)$, where n is the number of vertices and each edge is included with probability p, Erdős and Rényi [56] showed that there is a sharp threshold for connectivity. If p is slightly less than $\frac{\log n}{n}$, then the probability that $G(n,p)$ is connected tends to 0 as n increases, while if p is slightly greater than $\frac{\log n}{n}$, then the probability tends to 1. In graph theory, a monotone graph property is a property that only increases as more edges are added to a graph. The property of connectivity is an example of a monotone graph property. The result in [56] was later generalized by Friedgut and Kalai [66], who showed that every monotone graph property has a sharp threshold.

Let $\mu_p(\mathcal{P}_n)$ be the probability that a random graph $G \sim G(n,p)$ satisfies a property $\mathcal{P}_n$, and let $\mathsf{I}_{\mathcal{P}_n}$ denote the characteristic function of $\mathcal{P}_n$. According to the Margulis-Russo lemma [115, 147], $\frac{d(\mu_p(P_n))}{dp} = \mathsf{inf}(\mathsf{I}_{\mathcal{P}_n})$. Hence the property has sharp threshold if and only if the total influence of the characteristic function of $\mathcal{P}_n$ is large. Friedgut and Kalai were interested in identifying conditions that would lead to a large value of total influence. They proposed that a function with a significant symmetry is likely to have a spread-out Walsh spectrum, which would lead to a large total influence. This motivated them to formulate the Fourier-entropy/Influence conjecture [66], which is a fundamental and enduring open problem in the analysis of Boolean functions.

Conjecture 1 (***Fourier-entropy/Influence conjecture.***) *There exists a universal constant C such that for any integer $n \geq 1$ and for any n-variable Boolean function f, $H(f) \leq C \cdot \mathsf{inf}(f)$.*

In complexity theory, threshold phenomena are also observed. For example, if a random 3-SAT formula is selected with n variables and $m = cn$ clauses, then it is highly likely that the formula is satisfiable if $c < 4.2$, while it is highly likely to be unsatisfiable if $c > 4.2$ [52]. Kalai and Safra [95] have provided an insightful survey of such phenomena, demonstrating how the analysis of these phenomena relies on influences and Fourier techniques.

3.5 PCP and hardness of approximation

Fourier analysis has significant applications in the design and analysis of Probabilistically Checkable Proofs (PCP). The PCP theorem [3, 55], which is a well-known result in the field, states that a language is in NP if and only if it has witnesses that can be probabilistically verified using a constant number of queries to bits of the witness and $O(\log n)$ bits of randomness. The PCP theorem has important implications for the complexity of approximating optimization problems. In particular, it implies that many optimization problems are NP-hard to approximate within a certain factor, unless P=NP. This is known as the hardness of approximation result, and it shows that even if we cannot find the exact optimal solution to a problem efficiently, it may still be computationally difficult to find an approximate solution that is within a certain factor of the optimal. Håstad's 3-query PCP [82], which is one of the most efficient PCPs, uses Fourier analysis as its basis [52]. These PCPs can be used to demonstrate NP-hardness results for approximations of various optimization problems, including determining the maximum clique in a graph or the maximum number of satisfied clauses in a CNF formula.

The "Unique Games Conjecture" is another significant result in the field of hardness of approximation, introduced by Khot in 2002 [98]. This conjecture suggests that the computational complexity of determining the approximate value of a particular type of game, called a "unique game", is NP-hard. If this is assumed to be true, it can lead to almost optimal inapproximability outcomes for problems such as max-cut [100] and vertex cover [101]. Fourier techniques play a crucial role in the analysis of these results. For example, the "Majority Is Stablest" theorem [122] is a critical component of the max-cut result [52]. The theorem asserts that the "majority" function's noise stability is the highest among all balanced Boolean functions with low-influence variables. It should not be mistaken for the "Majority Is Least Stable Conjecture", which was proposed by Benjamini, Kalai, and Schramm in 1999 and specifically relates to the noise stability of majority within the category of "linear threshold functions". In Chapter 7, we provide evidence against this conjecture by presenting a counterexample that holds true for all values of $n \geq 5$.

3.6 Cryptography

In symmetric key (private key) cryptosystems, both the sender and receiver possess the same key, and the sender uses this key to encrypt the message, while the receiver decrypts the cipher using the same key. Since an attacker can intercept a portion of the cipher during transmission, their objective is to recover the key from the cipher. Boolean functions and S-boxes with certain cryptographic properties, such as correlation immunity, resiliency, balancedness, algebraic degree, non-linearity, strict avalanche criteria (SAC), and Propagation Characteristic (PC), are crucial in the design of secure symmetric key cryptographic systems. The Walsh/Fourier transform is a commonly used technique to efficiently and effectively analyze the cryptographic properties of Boolean functions. In this section, we will provide a brief explanation of why these cryptographic properties are necessary for a secure cryptographic design. We will also discuss the alternative spectrum characterization of each property. The content in this section is primarily sourced from [74, 36].

Correlation immunity and Resiliency. In stream ciphers, a popular type of running key generator involves combining the outputs of n binary linear feedback shift registers (LFSRs) using a Boolean combining function $f(x_1, \ldots, x_n)$, where x_i represents the output of the i^{th} LFSR. However, Siegenthaler showed that certain combining functions proposed in the literature are vulnerable to a "ciphertext-only correlation attack" [160]. To address this

vulnerability, Siegenthaler introduced the concept of m^{th}-order correlation immunity as a measure of a combining function's resistance to correlation attacks [159]. Specifically, an n-variable combining function f is said to be m^{th}-order correlation immune if the random variable $Z = f(X_1, X_2, \ldots, X_n)$ is statistically independent of every set of m random variables chosen from the balanced and independent binary random variables $X_1, X_2, \ldots, X_n$.

Xiao and Massey in [176] introduced a spectral characterization of correlation immune functions, which states that an n-variable function f is considered to be correlation immune of order t (t-CI) if $W_f(\boldsymbol{\alpha}) = 0$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ with $1 \leq \mathsf{wt}(\boldsymbol{\alpha}) \leq t$. Additionally, f is considered balanced if and only if $W_f(\mathbf{0}_n) = 0$, and a balanced t-CI function is referred to as t-resilient. Note that for t-CI (resp. t-resilient) functions, the algebraic degree d is bounded by $n-t$ (resp. $n-t-1$) [159]. Stream ciphers require a balanced function with high algebraic degree to protect against algebraic attacks [48].

However, it is worth mentioning that in contemporary cryptography, the combiner model is not commonly used. In this context, resiliency primarily plays a role concerning "guess and determine" attacks, which are a specific type of cryptographic attack. Additionally, correlation immunity is related to a particular kind of countermeasure against side-channel attacks, which are another class of attacks in cryptography.

Non-linearity. The nonlinearity of a Boolean function f, which is denoted as $\mathsf{nl}(f)$, is a crucial cryptographic property. This quantity is defined as the minimum Hamming distance between a Boolean function f and the set of all affine functions. When considering an S-box, the nonlinearity is determined by the minimum nonlinearity value among all non-constant component functions of that S-box.

For a Boolean function to be suitable for use in stream ciphers, it must possess high nonlinearity. If a function has low nonlinearity, it is vulnerable to linear approximation attacks, which involve approximating the combining function by a linear function. Therefore intuitively the requirements for a Boolean function in stream ciphers can be seen as the "opposite" of those in learning theory. Additionally, it is worth noting that high non-linearity is desirable for preventing fast correlation attacks [120, 36]. Matsui [117] introduced the linear cryptanalysis method for block ciphers, which involves approximating a linear combination of the coordinate functions of an S-box with a linear function of the input variables. In order for an S-box to resist linear cryptanalysis, it must have high nonlinearity. Therefore, for applications in symmetric ciphers, it is necessary to use functions, including both Boolean functions and S-boxes, that have high nonlinearity.

It is possible to measure the non-linearity of a Boolean function using its spectrum (see [111]) as follows: $\mathsf{nl}(f) = 2^{n-1} - 2^{n-1} \cdot \left|\max_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} W_f(\boldsymbol{\alpha})\right|$. Therefore the maximum nonlinearity achievable by an n-variable Boolean function is $2^{n-1} - 2^{\frac{n-2}{2}}$, which is reached by functions called "bent" and exists only when n is even [145]. However, bent functions cannot be used directly in cryptography as they are unbalanced. When n is odd, the maximum achievable nonlinearity of a Boolean function is not known, but functions achieving a nonlinearity of $2^{n-1} - 2^{\frac{n-1}{2}}$ can be constructed due to [135, 134]. To obtain balanced functions with high nonlinearity in either odd or even numbers of variables, "plateaued functions" were introduced by Zheng and Zhang in [179, 180], as a generalization of bent functions. Note that, it can be easily deduced from the definition of the non-linearity of S-boxes that the highest possible nonlinearity that can be attained by an (n, k) S-box is $2^{n-1} - 2^{\frac{n-2}{2}}$ When $k = 1$, S-boxes with perfect nonlinearity are equivalent to bent functions. These types of S-boxes can only exist when n is even and $k \leq \frac{n}{2}$ [126]. On the other hand, if n is odd and $k = n$, the highest attainable nonlinearity is $2^{n-1} - 2^{\frac{n-1}{2}}$. However, for odd n and $1 \leq k < n$, the maximum achievable nonlinearity is still an unsolved problem.

SAC and PC. The concept of SAC was first introduced by Webster and Tavares in their paper [173]. A Boolean function f with n variables satisfies SAC (Strict Avalanche Criterion) if, for any $\boldsymbol{\alpha}$ in $\mathbb{F}_2^n$ such that $\mathsf{wt}(\boldsymbol{\alpha}) = 1$, the function $f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \boldsymbol{\alpha})$ is balanced. Additionally, f satisfies SAC(l) if each subfunction derived from $f(x_1, \ldots, x_n)$ by fixing at most l input bits also satisfies SAC. Preneel et. al. [142] introduced the concept of PC (Propagation Characteristic), which is a more general version of SAC. An n-variable Boolean function f satisfies PC of degree l (i.e. PC(l)) if $f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \boldsymbol{\alpha})$ is balanced for any $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ such that $1 \leq \mathsf{wt}(\boldsymbol{\alpha}) \leq l$. An (n, k) S-box $\mathscr{G}$ is said to satisfy SAC(l) (resp. PC(l)) if all the non-constant component functions of $\mathscr{G}$ satisfy SAC(l) (resp. PC(l)).

The motivation behind SAC, PC(l), and SAC(l) is the following [74]. By satisfying SAC, the output of an S-box with an input difference of weight one will have a uniform distribution, reducing the propagation ratio and thus making the overall propagation ratio of the differential trail lower. The generalization of SAC is PC(l), where the input difference weight of any active S-box should be at least l to prevent uniform output distributions, which are highly unlikely for larger values of l.

Note that from the definition of PC(l) we can say an n-variable function to be satisfying PC of degree l (PC(l)) if $C_f(\boldsymbol{\alpha}) = 0$ for all $1 \leq \mathsf{wt}(\boldsymbol{\alpha}) \leq l$ [141]. Therefore, by Wiener-Khintchine formula (2.12), it is possible to obtain an alternate spectral characterisation of

PC(l).

In conclusion, achieving all desired properties simultaneously in cryptographic primitives is not feasible, and trade-offs between these properties are necessary. The importance of each property varies depending on the application, and cryptographic primitives should be designed accordingly. The Walsh transform plays a critical role in designing Boolean functions for stream ciphers or S-boxes for block ciphers since all of these properties can be characterized spectrally. This includes the design of non-linear correlation-immune/resilient functions [32, 43, 149, 153], non-linear resilient S-boxes [178, 133, 91, 77, 75], and nonlinear S-boxes satisfying higher-order SAC [76].

Chapter 4

Separation results for Boolean function classes

In this chapter we show (almost) separation between certain important classes of Boolean functions. What makes this chapter especially interesting is the idea that findings from one field can be used to prove results in the other, often with greater simplicity and understanding. Discovering these connections between fields, allowing us to apply findings from one field to prove results in the other, is a valuable aspect of our work.

To illustrate this point, consider the separation between bent and monotone functions. We present a shorter proof, compared to the one found in [37], by using results from theoretical computer science. Our approach revolves around the concept of total influence, which we use to show that the total influence of functions in one group is less than that of functions in the other group. Using this same approach, in this chapter we show (almost) separation of several classes of Boolean functions which have been studied in coding theory and cryptography from classes which have been studied in combinatorics and complexity theory.

4.1 Introduction

If the intersection of two classes of Boolean functions is empty, then the classes are disjoint. For two infinite classes of Boolean functions, we say that they are almost disjoint, if their intersection is a finite set. Being almost disjoint implies that there is a positive integer n_0 such that for any $n \geq n_0$, there is no n-variable Boolean function which belongs to both classes.

Our goal is to show that several pairs of classes of Boolean functions are almost disjoint. We use the notion of total influence [110] to show such separation. The technique that we use is to show that for sufficiently large n, the total influence of any n-variable Boolean function in one of the classes is less than the total influence of any n-variable Boolean function in the other class. It turns out that there are some known results on total influence which can be

effectively used with this technique.

The specific results that we obtain are the following. The class of Boolean function consisting of bent functions and functions satisfying SAC and PC is almost disjoint from the class of monotone functions; the class of functions which can be implemented using constant depth, polynomial size circuits; and the class of linear threshold functions. Similar separation results are obtained for the class of plateaued functions.

The separation of bent and monotone functions was conjectured in [39] and proved in [37]. A detailed analysis of the non-linearity of monotone functions has been performed in [35] a consequence of which is also a proof of the separation of bent and monotone functions. Our proof which is based on total influence is shorter than both the proofs in [37] and [35]. While [37, 35] had considered cryptographic properties of monotone functions, the classes of SAC, PC and plateaued functions were not considered in [37]. So, the separation results for these classes mentioned above are not present in [37].

4.2 Some Boolean function classes

In this chapter, we define the Boolean function classes necessary for our discussion. While some of these classes have been previously defined (refer to Chapter 3), our definitions here may differ, but they are still equivalent to the original definitions. We will specifically focus on describing the Boolean function classes relevant to cryptography in terms of the Walsh spectrum and auto-correlation function since that would be helpful for us to compute the total influence of a function belonging to that class.

An n-variable Boolean function f is said to be *monotone* if the following property holds. For $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n$, if $\mathbf{a} \leq \mathbf{b}$ (i.e., $a_i \leq b_i$, $i = 1, \ldots, n$), then $f(\mathbf{a}) \leq f(\mathbf{b})$. Let M denote the set of all monotone Boolean functions.

The following Boolean function classes have been studied in the context of coding theory and cryptography.

- For even n, an n-variable Boolean function f is said to be *bent* [145], if $W_f(\boldsymbol{\alpha}) = \pm\frac{1}{2^{n/2}}$, for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. Let B denote the set of all bent functions.

- An n-variable Boolean function f satisfies the *strict avalanche criterion* (SAC) [173], if for all $i \in [n]$, $C_f(\mathbf{e}_i) = 0$, where $\mathbf{e}_i$ represents a vector in $\mathbb{F}_2^n$ with its i^{th} element being 1 and all other elements being 0. Further, we say that an n-variable function

satisfies SAC of order k, $0 \leq k \leq n-2$, (written as SAC(k)) if by fixing any k of the n variables to arbitrary values in $\mathbb{F}_2$, the resulting function satisfies SAC. For $k \geq 0$, let S_k denote the set of all Boolean functions satisfying SAC(k), and define $\mathsf{S} = \cup_{k \geq 0} \mathsf{S}_k$.

- An n-variable Boolean function f satisfies *propagation characteristics* [142] of degree k, $1 \leq k \leq n$, (written as PC(k)) if $C_f(\boldsymbol{\alpha}) = 0$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ such that $1 \leq \mathsf{wt}(\boldsymbol{\alpha}) \leq k$. For $k \geq 1$, let PC_k denote the set of all Boolean functions satisfying PC(k), and define $\mathsf{PC} = \cup_{k \geq 1} \mathsf{PC}_k$. Note that for any k, $\mathsf{PC}_k \subseteq \mathsf{PC}_1$. Therefore, $\mathsf{PC} = \mathsf{PC}_1 = \mathsf{S}_0$.

- An n-variable Boolean function f is said to be *k-plateaued* [180], for $k \in \{0, \ldots, n\}$ and $n \equiv k \bmod 2$, if for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, $W_f(\boldsymbol{\alpha}) \in \left\{0, \pm \frac{1}{2^{(n-k)/2}}\right\}$. For $k \geq 0$, let PL_k be the set of all k-plateaued Boolean functions, and define $\mathsf{PL} = \cup_{k \geq 0} \mathsf{PL}_k$.

We next define some Boolean functions classes which have been studied in the context of complexity theory.

- A bounded-depth circuit [83] for n variables refers to a type of Boolean circuit that is constructed using AND and OR gates, and whose inputs consist of variables $x_1, \ldots, x_n$ as well as their negations. Fan-in to the gates is unbounded but depth is bounded by a constant. Without loss of generality, the circuit is leveled, where gates at level i have all their inputs from level $i-1$; all gates at the same level have the same type, i.e., all gates at a particular level are either AND or OR; and the types of gates alternate between AND and OR for successive levels. The depth of such a circuit is the number of levels that it has. The size of a circuit is the number of gates in it. If the size of a bounded depth circuit is bounded by a polynomial in n, then it is called an AC^0 circuit [110]. The set of all Boolean functions computable by AC^0 circuits of depth d is denoted by $\mathsf{AC}^0[d]$.

- A Boolean function f is said to be a *linear threshold function* [45], if there are real constants $w_0, w_1, \ldots, w_n$ such that for any $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$,

$$f(\mathbf{x}) = \frac{1 - \mathsf{sign}\left(w_0 + w_1 \cdot (-1)^{x_1} + \ldots + w_n \cdot (-1)^{x_n}\right)}{2},$$

  where $\mathsf{sign}(z) = 1$ if $z > 0$, and -1 if $z \leq 0$. Let LTF denote the set of all linear threshold functions. It is worthwhile to mention that it would be possible to translate the definition of linear threshold functions in terms of ANF [119].

Some known results. Below we collect together some relevant results on total influence that will be required for proving separation results. Some of these results were stated in terms of average sensitivity which is the same as total influence (see Lemma 6).

Fact 1 [27] : *For any n-variable function $f \in \mathsf{AC}^0[d]$, $\mathsf{inf}(f) = O((\log n)^{d-1})$.*

Fact 2 [177] : *For any non-constant n-variable* monotone *Boolean function f, $\mathsf{inf}(f) \leq \binom{n}{\lfloor n/2 \rfloor}\lceil n/2 \rceil / 2^{n-1}$. Since $\binom{n}{\lfloor n/2 \rfloor}\lceil n/2 \rceil / 2^{n-1} = \Theta(\sqrt{n})$, we have $\mathsf{inf}(f) = O(\sqrt{n})$.*

Fact 3 [54] : *For any n-variable Boolean function f in LTF, $\mathsf{inf}(f) \leq 2\sqrt{n}$.*

Fact 4 [68] : *For any n-variable Boolean function f in PL_k, $\mathsf{inf}(f) = \Omega(n-k)$.*

4.3 Separation Results

For $n \in \mathbb{N}$, let $\mathcal{C}_n^1$ and $\mathcal{C}_n^2$, be two subsets of the set of all n-variable Boolean functions. Define $\mathcal{C}^1 \triangleq \bigcup_{n\geq 1} \mathcal{C}_n^1$ and $\mathcal{C}^2 \triangleq \bigcup_{n\geq 1} \mathcal{C}_n^2$. Then $\mathcal{C}_1$ and $\mathcal{C}_2$ are two classes of Boolean functions. Suppose there exists a constant n_0, such that for all $n \geq n_0$, $\mathcal{C}_n^1 \bigcap \mathcal{C}_n^2 = \emptyset$. If $n_0 = 1$, then the classes $\mathcal{C}^1$ and $\mathcal{C}^2$ are disjoint. If $n_0 > 1$, then we say that the classes are n_0-disjoint. Note that if the classes $\mathcal{C}^1$ and $\mathcal{C}^2$ are n_0-disjoint, then their intersection is finite, i.e., they are almost disjoint.

Let $\mathcal{C}^1$ and $\mathcal{C}^2$ be two classes of Boolean functions. To show separation between $\mathcal{C}^1$ and $\mathcal{C}^2$ we use the following idea. Suppose it is possible to find a function $\mathcal{P}$ from the set of all Boolean functions to the reals and a positive integer n_0 such that for all $n \geq n_0$ and for all $f \in \mathcal{C}_n^1$ and $g \in \mathcal{C}_n^2$, $\mathcal{P}(f) < \mathcal{P}(g)$. Then, it follows that $\mathcal{C}_n^1$ and $\mathcal{C}_n^2$ are n_0-disjoint. We use total influence as the function $\mathcal{P}$. To do so, we need results on total influence for both classes. Results on total influence for some of the classes have been provided in Section 4.2. The following result provides the value of total influence for functions in $\mathsf{B} \cup \mathsf{PC} \cup \mathsf{PL}$.

Proposition 1 *If an n-variable Boolean function f is in $\mathsf{B} \cup \mathsf{S}$ then $\mathsf{inf}(f) = n/2$.*

Proof: For $f \in \mathsf{B}$, we have $W_f(\boldsymbol{\alpha}) = \pm\frac{1}{2^{n/2}}$, for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. Hence, using (2.19)

$$\mathsf{inf}(f) = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \mathsf{wt}(\boldsymbol{\alpha})(W_f(\boldsymbol{\alpha}))^2 = \frac{n}{2}.$$

According to the definition of SAC, it is evident that any function satisfying SAC(0) also satisfies the SAC. Suppose f satisfies SAC(k) for some $k \geq 1$. Then, it follows that f satisfies SAC(j) for $0 \leq j \leq k-1$ (see [50]). So, in particular, if f satisfies SAC(k) for some $k \geq 0$, then f must satisfy SAC. Now, for any Boolean function g, using (2.12) and (2.17), we can derive that $\mathsf{inf}_g(i)$ is equal to $\frac{1-C_g(e_i)}{2}$. Therefore, from the definition of SAC, we have $\mathsf{inf}_f(i) = \frac{1}{2}$ for all i, and so $\mathsf{inf}(f) = \frac{n}{2}$.

□

Theorem 8 *The following disjointness results hold for* $\mathsf{B} \cup \mathsf{S}$.

1. M *is* 4*-disjoint from* $\mathsf{B} \cup \mathsf{S}$.

2. LTF *is* 16*-disjoint from* $\mathsf{B} \cup \mathsf{S}$.

3. *Let* d *be any positive integer. Then there exists a positive integer* $\mathtt{n}_0$ *(depending on* d*) such that* $\mathsf{AC}^0[d]$ *is* $\mathtt{n}_0$*-disjoint from* $\mathsf{B} \cup \mathsf{S}$.

Proof: *Proof of the first point.* For any n-variable monotone Boolean function f, from Fact 2, we have $\mathsf{inf}(f) \leq \binom{n}{\lfloor n/2 \rfloor} \lceil n/2 \rceil / 2^{n-1}$. Since, for $n \geq 4$, we have $\binom{n}{\lfloor n/2 \rfloor} \lceil n/2 \rceil / 2^{n-1} < n/2$, it follows that for $n \geq 4$, $\mathsf{inf}(f) < n/2$. On the other hand, from Proposition 1 for any n-variable Boolean function in $\mathsf{B} \cup \mathsf{S}$,the total influence is equal to $\frac{n}{2}$. So, f cannot be in $\mathsf{B} \cup \mathsf{S}$.

Proof of the second point. Let f be any n-variable Boolean function in LTF. From Fact 3, $\mathsf{inf}(f) \leq 2\sqrt{n}$. Now, for $n > 16$, $2\sqrt{n} < \frac{n}{2}$. Therefore, using Proposition 1, we obtain the desired result.

Proof of the third point. Let $f \in \mathsf{AC}^0[d]$. From Fact 1, we have $\mathsf{inf}(f) = O((\log n)^{d-1})$. Consequently, there is a constant c and a positive integer n_1, such that $\mathsf{inf}(f) \leq c(\log n)^{d-1}$. Since d is fixed, there is a positive integer $\mathtt{n}_0$ such that $c(\log n)^{d-1} < n/2$ for all $n \geq \mathtt{n}_0$. So, for $n \geq \mathtt{n}_0$, $\mathsf{inf}(f) < n/2$. From Proposition 1, we have that for $n \geq \mathtt{n}_0$, f does not belong to $\mathsf{B} \cup \mathsf{S}$. □

The first point of Theorem 8 provides a shorter proof of the fact that no *monotone* function on $n \geq 4$ variables is bent, a result which was conjectured in [39] and originally proved in [37]. Note that $\mathsf{PC} \subseteq \mathsf{S}$. Therefore, the above disjoint result would also hold for functions belonging to the class PC. Furthermore, observe that an n-variable bent function satisfies

$\mathsf{PC}(n)$. So, showing that a Boolean function class $\mathcal{C}$ is $\mathtt{n}$-disjoint from the class S implies that $\mathcal{C}$ is also $\mathtt{n}$-disjoint from B.

The third point of Theorem 8 shows that in general bent functions and also functions satisfying and strict avalanche criteria cannot be realised using constant depth circuits. It has been shown in [110] that most of the spectral density of Boolean functions having constant depth circuits are on low weight Fourier coefficients. So, it is perhaps not surprising that such functions cannot be bent. We note, however, that there is no result in [110] from which it directly follows that class of Boolean functions with constant depth circuits is almost disjoint from either B, or S.

Corollary 3.5 of [73] states that if f is a linear threshold function, then $P_{\widehat{f}}(0)+P_{\widehat{f}}(1) \geq \frac{1}{2}$. Based on this result and utilizing Parseval's identity (see 2.8), it can be concluded that $(W_f(\boldsymbol{\alpha}))^2$ cannot be equal for every $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, indicating that f cannot be bent. Corollary 3.5 of [73] is a direct consequence of Theorem 3.3 of [73] whose proof is more involved than the simple technique of using total influence to separate the two classes that has been used in the present work.

The notion of linear threshold function has been extended to polynomial threshold function. An n-variable Boolean function f is said to be a degree d polynomial threshold function (PTF) [30] if there is a polynomial p of degree d, such that $f(x_1, x_2, \ldots, x_n) = \frac{1}{2}\left(1 - \mathsf{sign}\left(p\left((-1)^{x_1}, (-1)^{x_2}, \ldots, (-1)^{x_n}\right)\right)\right)$ for all $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$. It has been shown in [54] that if f is a degree d PTF, then $\mathsf{inf}(f) \leq 2^{O(d)} \cdot \log n \cdot n^{1-1/(4d+2)}$. Let PTF_d be the set of all degree d PTFs. In a manner similar to Theorem 8 that for $d = \log^c n$ with $c < 1/2$, the class $\mathsf{B} \cup \mathsf{S}$ is almost disjoint from PTF_d. It has been conjectured in [73] that if f is any n-variable degree d PTF, then $\mathsf{inf}(f) \leq d\sqrt{n}$. If the conjecture is true, it will show that $\mathsf{B} \cup \mathsf{S}$ is $4d^2$-disjoint from PTF_d.

Theorem 9 *Let k be a non-negative integer. The following disjointness results hold for PL_k.*

1. *There exists a positive integer $\mathtt{n}_0$ such that $\mathcal{M}$ and PL_k are $\mathtt{n}_0$-disjoint.*
2. *There exists a positive integer $\mathtt{n}_1$ such that LTF and PL_k are $\mathtt{n}_1$-disjoint.*
3. *Let d be any positive integer. There exists a positive integer $\mathtt{n}_2$ (depending on d) such that $\mathsf{AC}^0[d]$ and PL_k are $\mathtt{n}_2$-disjoint.*

Proof: *Proof of the first point.* Choose an $\varepsilon \in (1/2, 1)$ and let n be a positive integer satisfying $n - n^{\varepsilon} \geq k$. For any n-variable function f in PL_k, from Fact 4, $\mathsf{inf}(f) = \Omega(n-k) =$

$\Omega(n^{\varepsilon})$. So, there is a constant c_1 and an integer n_1, such that $\mathsf{inf}(f) \geq c_1 n^{\varepsilon}$ for all $n \geq n_1$. Let g be any n-variable function in M. From Fact 2, we have $\mathsf{inf}(g) = O(\sqrt{n})$. This implies that there is a constant c_2 and an integer n_2, such that $\mathsf{inf}(g) \leq c_2 n^{1/2}$ for all $n \geq n_2$. Since $\varepsilon > 1/2$ and c_1 and c_2 are constants, there is an integer $\mathtt{n}_0$ such that $c_1 n^{\varepsilon} > c_2 n^{1/2}$ for all $n \geq \mathtt{n}_0$. Note that $\mathtt{n}_0$ has to satisfy $\mathtt{n}_0 - \mathtt{n}_0^{\varepsilon} \geq k$. So, for $n \geq \mathtt{n}_0$, $\mathsf{inf}(f) \geq c_1 n^{\varepsilon} > c_2 n^{1/2} \geq \mathsf{inf}(g)$ which implies that f cannot be equal to g. Consequently, f cannot be equal to any function in M.

Proof of the second point. The proof is similar to the first point, with the only difference being that Fact 3 is used for the argument instead of Fact 2.

Proof of the third point. The proof is also similar to the first point, with the difference being that Fact 1 is used for the argument. □

We have already seen in Chapter 3 that it is possible to measure the non-linearity of a Boolean function using its spectrum as follows: $\mathsf{nl}(f) = 2^{n-1} - 2^{n-1} \cdot \left|\max_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} W_f(\boldsymbol{\alpha})\right|$. Therefore, for any function $f \in \mathsf{PL}_k$, $\mathsf{nl}(f) = 2^{n-1} - 2^{\frac{n}{2}+\frac{k}{2}-1}$. When we compare this with the non-linearity of monotone Boolean functions as provided in [35], we can readily observe that for even values of $n > 10$, there is a clear separation between the class of monotone functions and PL_1. However, for different values of k, this technique is unable to produce similar separation results. Additionally, when n is an odd number, we are unable to establish a clear separation between the class of monotone functions and PL_k for any $k \geq 1$, based on the non-linearity results available in [35].

Total influence can be used using to separate a few other classes of Boolean functions. We briefly mention these.

An n-variable Boolean function is said to have c-linearly high entropy [154] for real constant $c > 0$, if $H(f) \geq cn$. Let c-LHE denote the set of all Boolean functions having c-linearly high entropy. It has been shown in [154] that for $f \in \mathsf{LHE}_c$, with $c \in (0, \frac{1}{2})$, $H(f) \leq \frac{1+c}{h^{-1}(c^2)} \cdot \mathsf{inf}(f)$, where h^{-1} is the inverse of binary entropy function. Consequently, using Fact 1, it follows that for any positive integer d and $c \in (0, \frac{1}{2})$, LHE_c and $\mathsf{AC}^0[d]$ are almost disjoint.

The notion of random *linear threshold function* was considered in [40], where the parameters $w_0, w_1, \ldots, w_n$ are drawn independently from either the uniform distribution over $[-1, 1]$, or from the standard normal distribution. It has been shown [40] that for an n-variable random *linear threshold function* f, $\mathsf{inf}(f) = \Omega(\sqrt{n})$ with high probability. Combining with

Fact 1 we see that f is not in $\mathsf{AC}^0[d]$ with high probability, where d is any positive integer.

4.4 Conclusion

We have used total influence to separate classes of Boolean functions. In particular, we have shown separation of certain classes of Boolean functions of interest in coding theory and cryptography from classes of Boolean functions which have been considered in combinatorics and complexity theory.

Chapter 5

Influence of a set of variables on a Boolean function

Up to this point, we have explored the significance of the influence of a single variable in analyzing Boolean functions. However, the broader concept of the influence of a variable set on a Boolean function has been defined in four distinct ways in existing literature. In this chapter, we present a novel definition of variable set influence, which relies on the auto-correlation function, and develop its basic theory. Among the new results that we obtain are generalisations of the Poincaré inequality and the edge expansion property of the influence of a single variable. Further, we obtain new characterisations of resilient and bent functions using the notion of influence. We show that the previous definition of influence due to Fischer et al. (2002) and Blais (2009) is half the value of the auto-correlation based influence that we introduce. Regarding the other prior notions of influence, we make a detailed study of these and show that each of these definitions do not satisfy one or more desirable properties that a notion of influence may be expected to satisfy.

5.1 Introduction

As discussed in Chapter 3, the notion of variable influence on a Boolean function was initially introduced by Ben-Or and Linial in their work [12]. Subsequently, this concept has become central to the study of Boolean functions in various contexts. See [127] for a very comprehensive account of such applications. The notion of influence, however, has not received much attention in the context of cryptographic applications of Boolean functions. We know of only two works [69, 18] which studied influence in relation to cryptographic properties.

The notion of influence of a variable on a function has been extended to consider the influence of a set of variables on a function. We have been able to locate four different definitions of the influence of a set of variables on a Boolean function. The first definition appears in the work of Ben-Or and Linial [12] itself in 1989. A different definition due to Fischer et al. [62] appeared in 2002 and the same definition was considered in 2009 by

Blais [23]. A third definition was given by Gangopadhyay and Stănică [69] in 2014 and a fourth definition was given by Tal [164] in 2017. All of these definitions coincide with each other in the case of a single variable, but in the case of more than one variable, in general the values provided by the four definitions of influence are different.

The motivation of our work is to make a systematic and comprehensive study of the notion of influence of a set of variables on a Boolean function. To this end, we introduce a definition of influence based on the auto-correlation function, which is a very useful tool for analysing certain cryptographic properties of Boolean functions. Two Walsh transform based characterisations of influence are obtained and some basic intuitive properties are derived. Several results on the influence of a single variable are generalised. These include Poincaré inequality and edge expansion property of influence of a variable. In the context of cryptographic properties, we provide characterisations of resilient and bent functions using the notion of influence.

The definition of influence given in [62, 23] is shown to be half the value of the notion of influence that we introduce. We also argue that the definition of influence considered in [69] does not satisfy a basic desirable property, namely that the influence of a set of variables can be zero even if the function is not degenerate on these variables.

Next we define a quantity called pseudo-influence, obtain its Walsh transform based characterisation and derive certain basic properties. We show that the pseudo-influence does not satisfy some intuitive properties that one would expect a notion of influence to satisfy, which is why we call it pseudo-influence. From the Walsh transform based characterisation, it follows that the definition of influence considered by Tal [164] is the notion of pseudo-influence that we introduce. Our motivation for introducing pseudo-influence and analysing it is to show that the notion of influence considered in [164] is not satisfactory.

Lastly, we make a systematic study of the Ben-Or and Linial (BL) notion of influence [12]. We show that the BL notion of influence satisfies some desirable properties, but it does not satisfy sub-additivity. Further, we argue that compared to the auto-correlation based definition, the BL notion of influence is a more coarse measure.

Organization. Section 5.2 describes the previous definitions of influence of a set of variables. The definition of influence from auto-correlation is introduced in Section 5.3 and its Walsh transform based characterisations and basic properties are derived. The concept is further developed in several subsections. The path expansion property of influence is derived in Section 5.3.1, two probabilistic interpretations of influence are given in Section 5.3.2, the

relation of influence to juntas and cryptographic properties are described in Section 5.3.3 and 5.3.4 respectively, and a general form of the Fourier entropy/influence conjecture is mentioned in Section 5.3.5. The notion of pseudo-influence is defined in Section 5.4 and its properties as well as its relation to influence are studied. Section 5.5 makes a detailed investigation of the notion of influence introduced by Ben-Or and Linial and its relation to the auto-correlation based notion of influence. A discussion of the new results in this paper and their importance is given in Section 5.6. Finally, Section 5.7 concludes the paper.

5.2 Influence

Let us begin by revisiting the explanation of the impact of a single variable on a function, as discussed in Chapter 2. Suppose $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is an n-variable Boolean function with variables denoted as $X_1, X_2, \ldots, X_{n-1}, X_n$. For $i \in [n]$, the influence of X_i on f is denoted by $\mathsf{inf}_f(i)$ and is defined to be the probability (over a uniform random choice of $\mathbf{x} \in \mathbb{F}_2^n$) that $f(\mathbf{x})$ is not equal to $f(\mathbf{x} \oplus \mathbf{e}_i)$, i.e.,

$$\mathsf{inf}_f(i) \;=\; \Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x} \oplus \mathbf{e}_i)].$$

The total influence $\mathsf{inf}(f)$ of the individual variables is defined to be the sum of the influences of the individual variables, i.e. $\mathsf{inf}(f) = \sum_{i \in [n]} \mathsf{inf}_f(i)$.

Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq [n]$ with $t = \#T$. The influence of the set of variables indexed by T on f has been defined in the literature in four different ways. These definitions are given below.

Ben-Or and Linial [12]. The definition of influence introduced in [12] is the following.

$$\mathcal{I}_f(T) \;=\; \Pr_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left[f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_T) \text{ is not constant} \right]. \tag{5.1}$$

Fischer et al. [62] and Blais [23]. The same quantity has been defined in two different ways in Fischer et al. [62] and Blais [23]. In [62], this quantity was called 'variation' and in [23], it was termed 'influence'. Here we provide the formulation as given in [23]. For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, let $Z(T, \mathbf{x}, \mathbf{y})$ denote the vector $\mathbf{z} \in \mathbb{F}_2^n$, where $z_i = y_i$, if $i \in T$ and $z_i = x_i$

otherwise. The definition of influence given in [23] is the following.

$$I_f(T) \;=\; \Pr_{\mathbf{x},\mathbf{y}\in\mathbb{F}_2^n}\left[f(\mathbf{x}) \neq f(Z(T,\mathbf{x},\mathbf{y}))\right]. \tag{5.2}$$

Gangopadhyay and Stănică [69]. The definition of influence introduced in [69] is the following.

$$\mathcal{J}_f(T) \;=\; \Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq f(\mathbf{x}\oplus\chi_T)] = \frac{1}{2}\left(1 - C_f(\chi_T)\right). \tag{5.3}$$

Tal [164]. For $\boldsymbol{\beta} \in \mathbb{F}_2^t$, let $f_{\boldsymbol{\beta}}$ denote the function $f_{\mathbf{X}_T\leftarrow\boldsymbol{\beta}}$. Let $D_T f : \{0,1\}^{n-t} \to [-1,1]$ be defined as follows. For $\mathbf{y} \in F_2^{n-t}$, $(D_T f)(\mathbf{y}) = 1/2^t \times \sum_{\boldsymbol{\beta}\in\mathbb{F}_2^t}(-1)^{\mathsf{wt}(\boldsymbol{\beta})+f_{\boldsymbol{\beta}}(\mathbf{y})}$. The definition of influence given in [164] is the following.

$$J_f(T) \;=\; \mathop{\mathbb{E}}_{\mathbf{y}\in\mathbb{F}_2^{n-t}}\left[\left(D_T f(\mathbf{y})\right)^2\right]. \tag{5.4}$$

5.3 Influence from Auto-Correlation

The auto-correlation function is a very useful tool for expressing various properties of Boolean functions. We refer to [36] for the many uses of the auto-correlation function in the context of cryptographic properties of Boolean functions. Given an n-variable Boolean function f and $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, the value of the auto-correlation function C_f at $\boldsymbol{\alpha}$, i.e., $C_f(\boldsymbol{\alpha})$ is the number of places $f(\mathbf{X})$ and $f(\mathbf{X}\oplus\boldsymbol{\alpha})$ are equal minus the number of places they are unequal (normalised by 2^n). So the auto-correlation function at $\boldsymbol{\alpha}$ to some extent captures the effect on f of flipping all the bits in the support of $\boldsymbol{\alpha}$. This suggests that the auto-correlation function is an appropriate mechanism to capture the influence of a set of variables on a Boolean function. We note that for $i \in [n]$, $\mathsf{inf}_f(i)$ can be written as follows.

$$\mathsf{inf}_f(i) \;=\; \frac{1}{2}\left(1 - C_f(\mathbf{e}_i)\right) = 1 - \frac{1}{2}\left(C_f(\mathbf{0}) + C_f(\mathbf{e}_i)\right). \tag{5.5}$$

Let $f(X_1,\ldots,X_n)$ be an n-variable Boolean function and $\emptyset \neq T = \{i_1,\ldots,i_t\} \subseteq [n]$. We denote the influence of the set of variables $\{X_{i_1},\ldots,X_{i_t}\}$ corresponding to $T = \{i_1,\ldots,i_t\}$ on the Boolean function f by $\mathsf{inf}_f(T)$. Following the auto-correlation based expression of the influence of a single variable on a Boolean function given by (5.5), we put forward the

following definition of $\mathsf{inf}_f(T)$.

$$\mathsf{inf}_f(T) \;=\; 1 - \frac{1}{2^{\#T}} \left(\sum_{\boldsymbol{\alpha} \leq \chi_T} C_f(\boldsymbol{\alpha}) \right). \tag{5.6}$$

It is easy to note that for a singleton set $T = \{i\}$, $\mathsf{inf}_f(T) = \mathsf{inf}_f(i)$. Further, one may note that $\mathsf{inf}_f(T) = 2^{1-t} \times \sum_{S \subseteq T} \mathcal{J}_f(S)$.

Remark 3 *We note that* $\mathsf{inf}_f(T)$, $\mathcal{I}_f(T)$, $J_f(T)$ *and* $\mathcal{J}_f(T)$ *(defined in Section 5.2) agree with each other when* $\#T = 1$. *Also, we later show that* $I_f(T) = \mathsf{inf}_f(T)/2$.

It is perhaps not immediately obvious that the definition of influence given by (5.6) is appropriate. We later show in Theorem 14 that this definition satisfies a set of intuitive desiderata that any notion of influence may be expected to satisfy.

Let f be an n-variable function and t be an integer with $1 \leq t \leq n$. Then the t-influence of f is the total influence (scaled by $\binom{n}{t}$) obtained by summing the influence of every set of t variables on the function f, i.e.,

$$t\text{-}\mathsf{inf}(f) \;=\; \frac{\sum_{\{T \subseteq [n] : \#T = t\}} \mathsf{inf}_f(T)}{\binom{n}{t}}. \tag{5.7}$$

Note that 1-$\mathsf{inf}(f)$ is equal to $\mathsf{inf}(f)/n$, i.e., 1-$\mathsf{inf}(f)$ is the sum of the influences of the individual variables scaled by a factor of n.

The following result provides a characterisation of influence in terms of the Walsh transform.

Theorem 10 *Let* f *be an* n*-variable Boolean function and* $\emptyset \neq T \subseteq [n]$. *Then*

$$\mathsf{inf}_f(T) \;=\; \sum_{\{\mathbf{u} \in \mathbb{F}_2^n : \mathsf{supp}(\mathbf{u}) \cap T \neq \emptyset\}} \left(W_f(\mathbf{u})\right)^2. \tag{5.8}$$

Proof: Let $\#T = t$. Let E be the subspace $\{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{x} \leq \chi_{\overline{T}}\}$. Then $\#E = 2^{n-t}$ and $E^{\perp} = \{\mathbf{y} \in \mathbb{F}_2^n : \mathbf{y} \leq \chi_T\}$. Using (2.15), we obtain

$$\sum_{\mathbf{x} \leq \chi_{\overline{T}}} \left(W_f(\mathbf{x})\right)^2 \;=\; \frac{2^{n-t}}{2^n} \sum_{\mathbf{y} \leq \chi_T} C_f(\mathbf{y}) = \frac{1}{2^{\#T}} \sum_{\mathbf{y} \leq \chi_T} C_f(\mathbf{y}). \tag{5.9}$$

Using (5.9) with (5.6) and (2.8) we have

$$\mathsf{inf}_f(T) \;=\; 1 - \sum_{\mathbf{x} \leq \chi_{\overline{T}}} \left(W_f(\mathbf{x})\right)^2 = \sum_{\mathbf{w} \in \mathbb{F}_2^n} \left(W_f(\mathbf{w})\right)^2 - \sum_{\mathbf{x} \leq \chi_{\overline{T}}} \left(W_f(\mathbf{x})\right)^2 = \sum_{\mathbf{u} \not\leq \chi_{\overline{T}}} \left(W_f(\mathbf{u})\right)^2 .$$

The condition $\mathbf{u} \not\leq \chi_{\overline{T}}$ is equivalent to $\mathsf{supp}(\mathbf{u}) \cap T \neq \emptyset$. □

It is a well known result (see Page 52 of [127]) that for an n-variable Boolean function, the total influence of the individual variables, i.e., $\mathsf{inf}(f)$ is the expected value of a random variable which takes the value k with probability $P_{\widehat{f}}(k)$ for $k = 0, \ldots, n$. We generalise this result to the case of t-$\mathsf{inf}(f)$ for $t \geq 1$.

For positive integers n, t and k with, $1 \leq t \leq n$ and $0 \leq k \leq n$, fix a subset S of $[n]$ with $\#S = k$ and let $N_{n,t,k}$ be the number of subsets of $[n]$ of size t which contain at least one element of S. Then

$$N_{n,t,k} \;=\; \binom{n}{t} - \binom{n-k}{t} = \sum_{i=1}^{\min(k,t)} \binom{k}{i}\binom{n-k}{t-i}. \tag{5.10}$$

It follows that $N_{n,t,0} = 0$, $N_{n,t,k} = \binom{n}{t}$ for $n - t + 1 \leq k \leq n$, and $N_{n,1,k} = k$ for $k = 0, \ldots, n$.

Theorem 11 *Let f be an n-variable function and $t \in [n]$. Then*

$$t\text{-}\mathsf{inf}(f) \;=\; \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} N_{n,t,k}\; P_{\widehat{f}}(k) = \frac{1}{\binom{n}{t}} \mathbb{E}[Z], \tag{5.11}$$

where Z is the number of t-element subsets of $[n]$ which have a non-empty intersection with a set $S \subseteq [n]$ chosen with probability $(W_f(\chi_S))^2$.

Proof: We start with the proof of the first equality in (5.11). Consider $\mathbf{u} \in \mathbb{F}_2^n$ with $\#\mathsf{supp}(\mathbf{u}) = k$. For $1 \leq i \leq \min(k,t)$, the number of subsets T of $[n]$ of cardinality t whose intersection with $\mathsf{supp}(\mathbf{u})$ is of size i is $\binom{k}{i}\binom{n-k}{t-i}$. Summing over i provides the number of subsets T of $[n]$ of cardinality t with which $\mathsf{supp}(\mathbf{u})$ has a non-empty intersection. From (5.7) and Theorem 10, we have

$$\begin{aligned} t\text{-}\mathsf{inf}(f) \;&=\; \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} \sum_{\{\mathbf{u} \in \mathbb{F}_2^n : \mathsf{wt}(\mathbf{u}) = k\}} \sum_{i=1}^{\min(k,t)} \binom{k}{i}\binom{n-k}{t-i} \left(W_f(\mathbf{u})\right)^2 \\ &=\; \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} \sum_{i=1}^{\min(k,t)} \binom{k}{i}\binom{n-k}{t-i} \sum_{\{\mathbf{u} \in \mathbb{F}_2^n : \mathsf{wt}(\mathbf{u}) = k\}} \left(W_f(\mathbf{u})\right)^2 \end{aligned}$$

$$\begin{aligned} &= \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} N_{n,t,k} P_{\widehat{f}}(k) \\ &= \frac{1}{\binom{n}{t}} \mathbb{E}[Z]. \end{aligned}$$

The second equality in (5.11) follows from the observation that if $\#S = k$, then $Z = N_{n,t,k}$. □

Poincaré inequality (Theorem 7) states that the total influence of the individual variables, i.e., $\mathsf{inf}(f)$ is bounded below by $4\,\mathsf{Var}(f)$. We obtain a generalisation of this result as a corollary of Theorem 11.

Corollary 1 *Let f be an n-variable Boolean function and $t \in [n]$. Then*

$$t\text{-}\mathsf{inf}(f) \;\geq\; \frac{4t}{n}\,\mathsf{Var}(f). \tag{5.12}$$

Equality is achieved for $t = n$.

Proof: Note that for $0 \leq k \leq n$, $n - i + 1 > 0$ for $2 \leq i \leq k$ and so $1 - t/(n-i+1) < 1$ for $2 \leq i \leq k$. Using this, we have

$$\frac{\binom{n-k}{t}}{\binom{n}{t}} = \left(1 - \frac{t}{n}\right)\left(1 - \frac{t}{n-1}\right)\cdots\left(1 - \frac{t}{n-k+1}\right) \leq 1 - \frac{t}{n}.$$

It follows that for $k \in [n]$, $N_{n,t,k}/\binom{n}{t} \geq t/n$, where equality is achieved for $t = n$. So from (5.11),

$$t\text{-}\mathsf{inf}(f) \;\geq\; \frac{t}{n}\sum_{k=1}^{n} P_{\widehat{f}}(k) = \frac{t}{n}(1 - P_{\widehat{f}}(\mathbf{0}_n)) = \frac{4t}{n}\mathsf{Var}(f).$$

□

The Fourier/Walsh transform based expression for the total influence given by Theorem 11 is a useful result. Corollary 1 above provides a direct application of Theorem 11. In Theorem 15, proved later, we use the expression given by Theorem 11 to characterise the functions which achieve the maximum value of the total influence as resilient functions. In Theorem 16, also proved later, the expression is used to show that total influence is monotonic increasing in t. An additional application of Theorem 11 is given next.

Given an n-variable Boolean function f, we say that the Fourier spectrum of f is ϵ-

concentrated on coefficients of weights up to k if $\sum_{i\geq k} P_{\widehat{f}}(i) \leq \epsilon$. Lemma 11 shows that the Fourier spectrum of f is ϵ-concentrated on coefficients of weights up to k, where k is the least positive integer such that $k \geq n \times 1\text{-}\mathsf{inf}(f)/\epsilon$ and $1\text{-}\mathsf{inf}(f) \leq \epsilon \leq 1$. The following theorem generalises this result to arbitrary values of t.

Theorem 12 *For any n-variable Boolean function, $t \in [n]$ and $\epsilon \in [t\text{-}\mathsf{inf}(f), 1]$, the Fourier spectrum of f is ϵ-concentrated on coefficients of weights up to k_t, where k_t is the least positive integer such that*

$$k_t \;\geq\; t - 1 + (n - t + 1)\left(1 - (1 - x_t)^{1/t}\right), \tag{5.13}$$

and $x_t = \frac{t\text{-}\mathsf{inf}(f)}{\epsilon}$.

Proof: The condition given by (5.13) holds if and only if $(n - k_t) \leq (n - t + 1)(1 - x_t)^t$ which holds if and only if

$$\frac{(n-k_t)^t}{t!} \;\leq\; \frac{(n-t+1)^t}{t!}(1 - x_t). \tag{5.14}$$

Using the inequalities $\binom{n-k_t}{t} \leq (n-k_t)^t/t!$ and $(n-t+1)^t/t! \leq \binom{n}{t}$, from (5.14) we obtain $\binom{n-k_t}{t} \leq \binom{n}{t}(1-x_t)$, which holds if and only if

$$\binom{n}{t} - \binom{n-k_t}{t} \;\geq\; \binom{n}{t} x_t = \binom{n}{t}\frac{t\text{-}\mathsf{inf}(f)}{\epsilon}. \tag{5.15}$$

Let if possible that the Fourier transform of f is not ϵ-concentrated on coefficients of weights up to k_t. Then for k_t satisfying (5.15), we have $\sum_{k=k_t}^{n} P_{\widehat{f}}(k) > \epsilon$. From (5.11), we have

$$\begin{aligned}
t\text{-}\mathsf{inf}(f) &= \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} N_{n,t,k}\; P_{\widehat{f}}(k) \\
&= \frac{1}{\binom{n}{t}} \left(\sum_{k=1}^{k_t - 1} N_{n,t,k}\; P_{\widehat{f}}(k) + \sum_{k=k_t}^{n} N_{n,t,k}\; P_{\widehat{f}}(k) \right) \\
&\geq \frac{1}{\binom{n}{t}} \sum_{k=k_t}^{n} \left(\binom{n}{t} - \binom{n-k}{t} \right) P_{\widehat{f}}(k) \quad \text{(using (5.10))} \\
&\geq \frac{1}{\binom{n}{t}} \left(\binom{n}{t} - \binom{n-k_t}{t} \right) \sum_{k=k_t}^{n} P_{\widehat{f}}(k) \\
&> \frac{1}{\binom{n}{t}} \left(\binom{n}{t} - \binom{n-k_t}{t} \right) \epsilon \quad \text{(by assumption)}
\end{aligned}$$

$$\geq \quad t\text{-}\mathsf{inf}(f) \quad \text{(using (5.15))}.$$

This gives us the desired contradiction. □

An alternative Walsh transform based characterisation of influence is given by the following result.

Theorem 13 *Let f be an n-variable function and $\emptyset \neq T \subseteq [n]$. Then*

$$\mathsf{inf}_f(T) \quad = \quad 1 - \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2, \tag{5.16}$$

where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}$.

Proof: Let $\#T = t$. Let $E = \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{x} \leq \chi_{\overline{T}}\}$ and so $E^{\perp} = \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{x} \leq \chi_T\}$. Using (2.15) and (2.16) we have

$$\frac{1}{2^t} \sum_{\mathbf{u} \leq \chi_T} C_f(\mathbf{u}) \quad = \quad \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2.$$

Using the definition of influence given in (5.6), we obtain the required result. □

Remark 4 *Theorems 10 and 13 provide two different Walsh transform based characterisations of $\mathsf{inf}_f(T)$. The expression for $\mathsf{inf}_f(T)$ given by (5.16) can be computed in $O(2^n)$ time, while the expression given by (5.8) in general will require $O(n2^n)$ time using the fast Fourier transform algorithm to compute the required values of the Walsh transform.*

We obtain the following corollary of Theorem 13.

Corollary 2 *Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq [n]$. Then*

$$\mathsf{inf}_f(T) \quad = \quad \frac{1}{2^{n-2-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \mathsf{Var}(f_{\boldsymbol{\alpha}}) \tag{5.17}$$

where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}$.

Proof: Using (5.16), we have

$$\mathsf{inf}_f(T) \quad = \quad 1 - \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2$$

$$
\begin{aligned}
&= \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(1 - (W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2\right) \\
&= \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} 4\mathbb{E}\left(f_{\boldsymbol{\alpha}}\right)\left(1 - \mathbb{E}\left(f_{\boldsymbol{\alpha}}\right)\right) \\
&= \frac{1}{2^{n-2-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \mathsf{Var}(f_{\boldsymbol{\alpha}}). \qquad (5.18)
\end{aligned}
$$

□

One may consider some basic desiderata that any reasonable measure of influence should satisfy. Since we are considering normalised measures, the value of influence should be in the set $[0, 1]$ and it should take the value 0 if and only if the function is degenerate on the set of variables. Further, by expanding a set of variables, the value of influence should not decrease, i.e. influence should be monotonic non-decreasing. Also, sub-additivity is a desirable property. The following result shows these properties for $\mathsf{inf}_f(T)$ and also characterises the condition under which $\mathsf{inf}_f(T)$ takes its maximum value 1.

Theorem 14 *Let f be an n-variable Boolean function and $\emptyset \neq T, S \subseteq [n]$. Then*

1. $0 \leq \mathsf{inf}_f(T) \leq 1$.
2. $\mathsf{inf}_f(T) = 0$ *if and only if the function f is degenerate on the variables indexed by T.*
3. $\mathsf{inf}_f(T) = 1$ *if and only if $f_{\boldsymbol{\alpha}}$ is balanced for each $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}$.*
4. $\mathsf{inf}_f(S \cup T) \geq \mathsf{inf}_f(S)$.
5. $\mathsf{inf}_f(S \cup T) = \mathsf{inf}_f(S) + \mathsf{inf}_f(T) - \sum_{\mathbf{u} \in \mathcal{U}} \left(W_f(\mathbf{u})\right)^2$, *where* $\mathcal{U} = \{\mathbf{u} \in \mathbb{F}_2^n : \mathsf{supp}(\mathbf{u}) \cap S \neq \emptyset \neq \mathsf{supp}(\mathbf{u}) \cap T\}$. *Consequently,* $\mathsf{inf}_f(S \cup T) \leq \mathsf{inf}_f(S) + \mathsf{inf}_f(T)$ *(i.e.,* $\mathsf{inf}_f(T)$ *satisfies sub-additivity).*

Proof: The first point follows from Theorem 10 and Parseval's theorem. The fourth and fifth points also follow from Theorem 10. The third point follows from Theorem 13.

Consider the second point. From (5.16), $\mathsf{inf}_f(T) = 0$ if and only if $\sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 = 2^{n-t}$. Since $\left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 \leq 1$, it follows that $\sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 = 2^{n-t}$ if and only if $\left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 = 1$ (equivalently, $f_{\boldsymbol{\alpha}}$ is constant) for all $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$. The last condition is equivalent to the statement that f is degenerate on the set of variables indexed by T. □

Remark 5 *For the Gangopadhyay and Stănică notion of influence $\mathcal{J}_f(T)$ (see 5.3) the second point of Theorem 14 does not hold. It is possible that f is not degenerate on the variables indexed by T, yet $\mathcal{J}_f(T) = 0$. For example, let $f(X_1, X_2, X_3, X_4) = (1 \oplus X_1)X_2(X_3 \oplus X_4)$ and $T = \{3, 4\}$. Then it may be checked that $\mathcal{J}_f(T) = 0$, but f is not degenerate on the set of variables $\{X_3, X_4\}$ as $f(0, 1, 0, 0) = 0 \neq f(0, 1, 0, 1)$.*

If a function is not degenerate on the set of variables indexed by T, then these variables have an effect on value of f. Any reasonable measure of influence should ensure that if f is not degenerate on a set of variables, then the value of the measure for this set of variables is positive. Since this condition does not hold for $\mathcal{J}_f(T)$, this measure cannot be considered to be a satisfactory measure of influence of a set of variables.

Theorem 15 *Let f be an n-variable Boolean function and t be an integer with $1 \leq t \leq n$.*

1. *t-$\mathsf{inf}(f)$ takes its maximum value 1 if and only if f is $(n-t)$-resilient.*
2. *t-$\mathsf{inf}(f)$ takes its minimum value 0 if and only if f is a constant function.*

Proof: From (5.11) and recalling that $N_{n,t,0} = 0$ and $N_{n,t,k} = \binom{n}{t}$ for $n - t + 1 \leq k \leq n$, we have

$$\begin{aligned} t\text{-}\mathsf{inf}(f) &= \frac{1}{\binom{n}{t}} \sum_{k=1}^{n} N_{n,t,k} P_{\widehat{f}}(k) \\ &= \frac{1}{\binom{n}{t}} \left(\sum_{k=0}^{n-t} \left(\binom{n}{t} - \binom{n-k}{t} \right) + \sum_{k=n-t+1}^{n} \binom{n}{t} \right) P_{\widehat{f}}(k) \\ &= \frac{1}{\binom{n}{t}} \left(\sum_{k=0}^{n} \binom{n}{t} - \sum_{k=0}^{n-t} \binom{n-k}{t} \right) P_{\widehat{f}}(k) \\ &= 1 - \frac{1}{\binom{n}{t}} \sum_{k=0}^{n-t} \binom{n-k}{t} P_{\widehat{f}}(k). \end{aligned} \tag{5.19}$$

From (5.19), t-$\mathsf{inf}(f)$ takes its maximum value of 1 if and only if $\sum_{k=0}^{n-t} \binom{n-k}{t} P_{\widehat{f}}(k) = 0$ which holds if and only if $P_{\widehat{f}}(k) = 0$ for $k = 0, \ldots, n-t$, i.e., if and only if f is $(n-t)$-resilient. This shows the first point.

For the second point, from (5.19), t-$\mathsf{inf}(f) = 0$ if and only if

$$\binom{n}{t} P_{\widehat{f}}(0) + \binom{n-1}{t} P_{\widehat{f}}(1) + \cdots + \binom{t}{t} P_{\widehat{f}}(t) = \binom{n}{t}. \tag{5.20}$$

If f is a constant function, then $P_{\widehat{f}}(0) = 1$ and $P_{\widehat{f}}(k) = 0$ for $k \in [n]$. So (5.20) holds. On the other hand, if f is not a constant function, then $P_{\widehat{f}}(0) < 1$. In this case,

$$\begin{aligned}
&\binom{n}{t}P_{\widehat{f}}(0) + \binom{n-1}{t}P_{\widehat{f}}(1) + \cdots + \binom{t}{t}P_{\widehat{f}}(t) \\
\leq\ &\binom{n}{t}P_{\widehat{f}}(0) + \binom{n-1}{t}(P_{\widehat{f}}(1) + \cdots + P_{\widehat{f}}(n)) \\
=\ &\binom{n}{t}P_{\widehat{f}}(0) + \binom{n-1}{t}(1 - P_{\widehat{f}}(0)) < \binom{n}{t}.
\end{aligned}$$

□

The next result shows that as t increases, the value of t-$\mathsf{inf}(f)$ is non-decreasing.

Theorem 16 *Let f be an n-variable Boolean function. For $t \in [n]$, t-$\mathsf{inf}(f)$ increases monotonically with t.*

Proof: For $t \in [n-1]$, the following calculations show that t-$\mathsf{inf}(f)$ is at most $(t+1)$-$\mathsf{inf}(f)$.

$$\begin{aligned}
&t\text{-}\mathsf{inf}(f) \leq (t+1)\text{-}\mathsf{inf}(f) \\
\iff\ &1 - \sum_{k=0}^{n-t} \frac{\binom{n-k}{t}}{\binom{n}{t}} P_{\widehat{f}}(k) \leq 1 - \sum_{k=0}^{n-t-1} \frac{\binom{n-k}{t+1}}{\binom{n}{t+1}} P_{\widehat{f}}(k) \\
\iff\ &\sum_{k=0}^{n-t} \frac{\binom{n-k}{t}}{\binom{n}{t}} P_{\widehat{f}}(k) \geq \sum_{k=0}^{n-t-1} \frac{\binom{n-k}{t+1}}{\binom{n}{t+1}} P_{\widehat{f}}(k) \\
\iff\ &\frac{1}{\binom{n}{t}} P_{\widehat{f}}(n-t) + \sum_{k=0}^{n-t-1} \left(\frac{\binom{n-k}{t}}{\binom{n}{t}} - \frac{\binom{n-k}{t+1}}{\binom{n}{t+1}} \right) P_{\widehat{f}}(k) \geq 0 \\
\iff\ &\frac{1}{\binom{n}{t}} P_{\widehat{f}}(n-t) + \sum_{k=0}^{n-t-1} \left(\frac{(n-k)!(n-t-1)!}{n!(n-k-t-1)!} \frac{k}{n-t-k} \right) P_{\widehat{f}}(k) \geq 0. \qquad (5.21)
\end{aligned}$$

For k in the range 0 to $n-t-1$, it follows that $k/(n-t-k) \geq 0$. So the relation in (5.21) holds showing that t-$\mathsf{inf}(f) \leq (t+1)$-$\mathsf{inf}(f)$.

□

5.3.1 Geometric Interpretation

Let H_n be the n-dimensional hypercube, i.e., H_n is a graph whose vertex set is $\mathbb{F}_2^n$ and two vertices $\mathbf{u}$ and $\mathbf{v}$ are connected by an edge if $\mathbf{v}$ can be obtained from $\mathbf{u}$ by flipping one of

the bits of $\mathbf{u}$, i.e., if $\mathsf{wt}(\mathbf{u} \oplus \mathbf{v}) = 1$. Let A be a subset of the vertices of H_n and $\overline{A} = \mathbb{F}_2^n \setminus A$. Let $e(A, \overline{A})$ be the number of edges between A and $\overline{A}$. Suppose f is an n-variable Boolean function such that $\mathsf{supp}(f) = A$. It is known that $\mathsf{inf}(f) = e(A, \overline{A})/2^{n-1}$ (see [94] and Page 52 of [127]). This relation is called the edge expansion property of influence. In this section, we obtain a general form of this relation for t-$\mathsf{inf}(f)$.

Suppose $\mathbf{u}$ is a vertex of H_n and $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ with $T = \mathsf{supp}(\boldsymbol{\alpha})$ and $t = \#T$. Let $\mathbf{v} = \mathbf{u} \oplus \boldsymbol{\alpha}$. Then $\mathbf{v}$ is obtained from $\mathbf{u}$ by flipping the bits of $\mathbf{u}$ which are indexed by T. Since these bits can be flipped in any order, there are a total of $t!$ paths of length t in H_n between $\mathbf{u}$ and $\mathbf{v}$.

Let A be a subset of H_n and f be an n-variable Boolean function such that $\mathsf{supp}(f) = A$. For $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, let $n_{\boldsymbol{\alpha}}$ be the number of paths between A and $\overline{A}$ such that the two ends $\mathbf{u}$ and $\mathbf{v}$ of any such path satisfy $\mathbf{u} \oplus \mathbf{v} = \boldsymbol{\alpha}$. The following result relates $n_{\boldsymbol{\alpha}}$ to the autocorrelation of f at $\boldsymbol{\alpha}$.

Proposition 2 $C_f(\boldsymbol{\alpha}) = 1 - \dfrac{n_{\boldsymbol{\alpha}}}{(\mathsf{wt}(\boldsymbol{\alpha}))! 2^{n-2}}$.

Proof: Let $x_{\boldsymbol{\alpha}} = \#\{(\mathbf{u}, \mathbf{v}) : \mathbf{u} \in A,\ \mathbf{v} \in \overline{A},\ \mathbf{u} \oplus \mathbf{v} = \boldsymbol{\alpha}\}$. Then

$$n_{\boldsymbol{\alpha}} \quad = \quad (\mathsf{wt}(\boldsymbol{\alpha}))! x_{\boldsymbol{\alpha}}. \tag{5.22}$$

Note that $x_{\boldsymbol{\alpha}} = \#\{\mathbf{u} \in \mathbb{F}_2^n : f(\mathbf{u}) = 1 \text{ and } f(\mathbf{u} \oplus \boldsymbol{\alpha}) = 0\}$. Let $g(\mathbf{X}) = f(\mathbf{X}) \oplus f(\mathbf{X} \oplus \boldsymbol{\alpha})$. Then

$$\begin{aligned} \mathsf{wt}(g) &= \#\{\mathbf{u} \in \mathbb{F}_2^n : \text{ either } f(\mathbf{u}) = 1 \text{ and } f(\mathbf{u} \oplus \boldsymbol{\alpha}) = 0, \text{ or } f(\mathbf{u}) = 0 \text{ and } f(\mathbf{u} \oplus \boldsymbol{\alpha}) = 1\} \\ &= 2\#\{\mathbf{u} \in \mathbb{F}_2^n : f(\mathbf{u}) = 1 \text{ and } f(\mathbf{u} \oplus \boldsymbol{\alpha}) = 0\} \\ &= 2x_{\boldsymbol{\alpha}}. \end{aligned} \tag{5.23}$$

From the definition of $C_f(\boldsymbol{\alpha})$ given in (2.12), it follows that $\mathsf{wt}(g) = 2^{n-1}(1 - C_f(\boldsymbol{\alpha}))$ which combined with (5.22) and (5.23) shows the result. □

Remark 6 *Proposition 2 connects auto-correlation to number of paths and consequently provides a geometric interpretation of the auto-correlation function. Combining Proposition 2 with* (2.13)*, we obtain*

$$(W_f(\boldsymbol{\beta}))^2 \quad = \quad \Delta_{\boldsymbol{\beta}} - \frac{1}{2^{2n-2}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} (-1)^{\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle} \frac{n_{\boldsymbol{\alpha}}}{(\mathsf{wt}(\boldsymbol{\alpha}))!},$$

where $\Delta_{\boldsymbol{\beta}} = 1$ *if* $\boldsymbol{\beta} = \mathbf{0}_n$ *and* 0 *otherwise. This provides a geometric interpretation of*

the Walsh transform. To the best of our knowledge, these geometric interpretations of the auto-correlation function and the Walsh transform do not appear earlier in the literature.

Now we are ready to state the path expansion property of t-$\mathsf{inf}(f)$.

Theorem 17 *Let f be an n-variable Boolean function and $t \in [n]$. Then*

$$t\text{-}\mathsf{inf}(f) = 1 - \frac{1}{2^{n+t-2}\binom{n}{t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \binom{n-\mathsf{wt}(\boldsymbol{\alpha})}{t-\mathsf{wt}(\boldsymbol{\alpha})} \left(2^{n-2} - \frac{n_{\boldsymbol{\alpha}}}{(\mathsf{wt}(\boldsymbol{\alpha}))!}\right). \tag{5.24}$$

Proof: Using Proposition 2 in the definition of $\mathsf{inf}_T(f)$ given by (5.6), we have

$$\begin{aligned}
\mathsf{inf}_T(f) &= 1 - \frac{1}{2^t}\left(\sum_{\boldsymbol{\alpha} \le \chi_T} C_f(\boldsymbol{\alpha})\right) \\
&= 1 - \frac{1}{2^t}\sum_{k=0}^{t}\left(\sum_{\boldsymbol{\alpha} \le \chi_T, \mathsf{wt}(\boldsymbol{\alpha})=k} C_f(\boldsymbol{\alpha})\right) \\
&= 1 - \frac{1}{2^t}\sum_{k=0}^{t}\left(\sum_{\boldsymbol{\alpha} \le \chi_T, \mathsf{wt}(\boldsymbol{\alpha})=k} \left(1 - \frac{n_{\boldsymbol{\alpha}}}{k!2^{n-2}}\right)\right).
\end{aligned} \tag{5.25}$$

For $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ with $\mathsf{wt}(\boldsymbol{\alpha}) = k$, there are exactly $\binom{n-k}{t-k}$ subsets T of $[n]$ such that $\alpha \le \chi_T$. Using this observation, we have

$$\begin{aligned}
t\text{-}\mathsf{inf}(f) &= \frac{1}{\binom{n}{t}} \sum_{T \subseteq [n], \#T=t} \mathsf{inf}_f(T) \\
&= 1 - \frac{1}{2^t\binom{n}{t}}\sum_{k=0}^{t}\left(\sum_{\{\boldsymbol{\alpha}:\mathsf{wt}(\boldsymbol{\alpha})=k\}} \binom{n-k}{t-k}\left(1 - \frac{n_{\boldsymbol{\alpha}}}{k!2^{n-2}}\right)\right) \\
&= 1 - \frac{1}{2^{n+t-2}\binom{n}{t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^n} \binom{n-\mathsf{wt}(\boldsymbol{\alpha})}{t-\mathsf{wt}(\boldsymbol{\alpha})} \left(2^{n-2} - \frac{n_{\boldsymbol{\alpha}}}{(\mathsf{wt}(\boldsymbol{\alpha}))!}\right).
\end{aligned}$$

□

Putting $t = 1$ in (5.24), we obtain $1\text{-}\mathsf{inf}(f) = \sum_{i \in [n]} n_{\mathbf{e}_i}/(n2^{n-1}) = e(A, \overline{A})/(n2^{n-1})$ which is the previously mentioned edge expansion property for $\mathsf{inf}(f)$ scaled by a factor of n.

5.3.2 Probabilistic Interpretation

We have defined the influence of a set of variables using the auto-correlation function. In this section, we provide two probabilistic interpretations of the influence.

Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq [n]$, with $\#T = t$. We define the following probability

$$\mu_f(T) \;=\; \Pr_{\boldsymbol{\alpha} \leq \chi_T, \mathbf{u} \in \mathbb{F}_2^n}[f(\mathbf{u}) \neq f(\mathbf{u} \oplus \boldsymbol{\alpha})]. \tag{5.26}$$

In (5.26), $\boldsymbol{\alpha}$ is required to be chosen uniformly at random from the set $\{\mathbf{x} : \mathbf{x} \leq \chi_T\}$. This is achieved by fixing the positions of $\boldsymbol{\alpha}$ corresponding to the elements of $\overline{T}$ to be 0, and choosing the bits of $\boldsymbol{\alpha}$ corresponding to the positions in T uniformly at random.

The definition of influence given by Fischer et al. [62] and Blais [23] is $I_f(T)$ and is given by (5.2). This definition is made in terms of the function $Z(T, \mathbf{x}, \mathbf{y})$. For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, both $\mathbf{x}$ and $Z(T, \mathbf{x}, \mathbf{y})$ agree on the bits indexed by $\overline{T}$. In particular, the bits of $\mathbf{y}$ indexed by $\overline{T}$ do not play any role in the probability $\Pr_{\mathbf{x},\mathbf{y} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq f(Z(T, \mathbf{x}, \mathbf{y}))]$. So this probability is the same as the probability of the event arising from choosing $\boldsymbol{\beta}$ uniformly at random from $\mathbb{F}_2^{n-t}$, choosing $\mathbf{w}$ and $\mathbf{z}$ independently and uniformly from $\mathbb{F}_2^t$ and considering $f_{\boldsymbol{\beta}}(\mathbf{w}) \neq f_{\boldsymbol{\beta}}(\mathbf{z})$. This shows that

$$I_f(T) \;=\; \Pr_{\boldsymbol{\beta} \in \mathbb{F}_2^{n-t}, \mathbf{w}, \mathbf{z} \in \mathbb{F}_2^t}[f_{\boldsymbol{\beta}}(\mathbf{w}) \neq f_{\boldsymbol{\beta}}(\mathbf{z})]. \tag{5.27}$$

where $f_{\boldsymbol{\beta}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\beta}}$.

The following result relates the above two probabilities to influence.

Theorem 18 *Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq [n]$. Then $\mu_f(T) = I_f(T) = \mathsf{inf}_f(T)/2$.*

Proof: We separately show that $\mu_f(T) = \mathsf{inf}_f(T)/2$ and $I_f(T) = \mathsf{inf}_f(T)/2$. Let $t = \#T$.

$$\begin{aligned} \mu_f(T) \;&=\; \frac{1}{2^t} \sum_{\boldsymbol{\alpha} \leq \chi_T} \Pr_{\mathbf{u} \in \mathbb{F}_2^n}[f(\mathbf{u}) \neq f(\mathbf{u} \oplus \boldsymbol{\alpha})] \\ &=\; \frac{1}{2^t} \sum_{\boldsymbol{\alpha} \leq \chi_T} \frac{1 - C_f(\alpha)}{2} \quad \text{(using (2.12))} \end{aligned}$$

$$\begin{aligned} &= \frac{1}{2}\left(1 - \frac{1}{2^t}\sum_{\boldsymbol{\alpha}\leq\chi_T} C_f(\alpha)\right) \\ &= \frac{\mathsf{inf}_f(T)}{2}. \end{aligned} \tag{5.28}$$

$$\begin{aligned} I_f(T) &= \frac{1}{2^{n-t}}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^{n-t}} \Pr_{\mathbf{w},\mathbf{z}\in\mathbb{F}_2^t}[f_{\boldsymbol{\beta}}(\mathbf{w}) \neq f_{\boldsymbol{\beta}}(\mathbf{z})] \\ &= \frac{1}{2^{n-t}}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^{n-t}} 2\times\frac{\mathsf{wt}(f_{\boldsymbol{\beta}})}{2^t}\left(1-\frac{\mathsf{wt}(f_{\boldsymbol{\beta}})}{2^t}\right) \\ &= \frac{1}{2^{n-1-t}}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^{n-t}} \mathbb{E}(f_{\boldsymbol{\beta}})(1-\mathbb{E}(f_{\boldsymbol{\beta}})) \\ &= \frac{1}{2^{n-1-t}}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^{n-t}} \mathsf{Var}(f_{\boldsymbol{\beta}}) \\ &= \frac{\mathsf{inf}_f(T)}{2} \quad \text{(from (5.17))}. \end{aligned}$$

□

Using the third point of Theorem 14, a consequence of Theorem 18 is that both the probabilities $\mu_f(T)$ and $I_f(T)$ are at most $1/2$.

Remark 7 *From Theorem 18, it follows that* $I_f(T) = \mathsf{inf}_f(T)/2$. *Some of the results for* $\mathsf{inf}_T(f)$ *that we have proved have been obtained for* $I_f(T)$ *in [62, 23]. In particular, it has been shown that* $I_f(T)$ *is equal to half the right hand side of* (5.8) *using a somewhat long proof which is different from the one that we given. Since we defined influence using the auto-correlation function, we were able to use known results on Walsh transform which make our proof simpler. Further, it has been proved in [62, 23] that* $I_f(T) \leq I_f(S\cup T) \leq I_f(S)+I_f(T)$, *i.e., monotonicity and sub-additivity properties hold for* I_f. *These properties for* $\mathsf{inf}_f(T)$ *are covered by Points 4 and 5 of Theorem 14.*

5.3.3 Juntas

The total influence of the individual variable, i.e. $\mathsf{inf}(f)$, for an s-junta f is known to be at most s. The following result generalises this to provide an upper bound on t-$\mathsf{inf}(f)$ for an s-junta.

Proposition 3 *Let f be an n-variable function which is an s-junta for some $s \in [n]$. For $t \in [n]$, t-$\mathsf{inf}(f) \leq 1 - \binom{n-s}{t}/\binom{n}{t}$.*

Proof: Let $T \subseteq [n]$ with $\#T = t$. Since f is an s-junta, there is a subset $S \subseteq [n]$, with $\#S \leq s$ such that f is degenerate on the variables indexed by $\overline{S}$. So $\mathsf{inf}_f(T) = 0$ if T is a subset of $\overline{S}$. This means that for $\binom{n-s}{t}$ possible subsets T, $\mathsf{inf}_f(T) = 0$. For the other $\binom{n}{t} - \binom{n-s}{t}$ possible subsets T, $\mathsf{inf}_f(T) \leq 1$. The result now follows from the definition of t-$\mathsf{inf}(f)$ given in (5.7). □

For $t = 1$, the upper bound on 1-$\mathsf{inf}(f)$ given by Proposition 3 is s/n which is a scaled version of the bound $\mathsf{inf}(f) \leq s$. Note that the upper bound on t-$\mathsf{inf}(f)$ increases as t increases and reaches 1 for $t > n - s$.

An n-variable Boolean function f is said to be ϵ-far from being a s-junta if for every n-variable s-junta g, $\Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq g(\mathbf{x})] \geq \epsilon$. It was proved in [23] that if f is ϵ-far from being an s-junta, then for any set $S \subseteq [n]$ with $\#S \leq s$, $I_f(\overline{S}) \geq \epsilon$. The following result provides an equivalent statement for $\mathsf{inf}_f(\overline{S})$. The reason for stating the result in the present work is that our proof is simpler than that in [23].

Proposition 4 *If an n-variable Boolean function f is ϵ-far from being an s-junta, then for any set $S \subseteq [n]$ with $\#S \leq s$, $\mathsf{inf}_f(\overline{S}) \geq 2\epsilon$.*

Proof: Among all the s-juntas on the variables indexed by S, let g be the closest s-junta to f. For $\boldsymbol{\alpha} \in \mathbb{F}_2^s$, let $f_{\boldsymbol{\alpha}} = f_{\mathbf{X}_S \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_{\overline{S}})$ and $g_{\boldsymbol{\alpha}} = g_{\mathbf{X}_S \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_{\overline{S}})$ be functions on $(n-s)$-variables. Since g is a junta on S, it is degenerate on all variables indexed by $\overline{S}$. So $g_{\boldsymbol{\alpha}}$ is a constant function for all $\boldsymbol{\alpha} \in \mathbb{F}_2^s$. Since among all the juntas on the variables indexed by S, g is the closest s-junta to f, it follows that for each $\boldsymbol{\alpha} \in \mathbb{F}_2^s$, $g_{\boldsymbol{\alpha}}$ is either the constant function 0 or the constant function 1 according as $\mathsf{wt}(f_{\boldsymbol{\alpha}}) \leq 2^{n-s-1}$ (i.e. $\mathbb{E}(f_{\boldsymbol{\alpha}}) \leq 1/2$) or $\mathsf{wt}(f_{\boldsymbol{\alpha}}) > 2^{n-s-1}$ (i.e. $\mathbb{E}(f_{\boldsymbol{\alpha}}) > 1/2$) respectively. So

$$\begin{aligned} \Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq g(\mathbf{x})] &= \frac{\sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^s} \mathsf{wt}(f_{\boldsymbol{\alpha}} \oplus g_{\boldsymbol{\alpha}})}{2^n} \\ &= \frac{1}{2^s} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^s} \min\left(\mathbb{E}(f_{\boldsymbol{\alpha}}), 1 - \mathbb{E}(f_{\boldsymbol{\alpha}})\right). \end{aligned} \tag{5.29}$$

Since f is ϵ-far from being an s-junta, it follows that $\epsilon \leq \Pr_{\mathbf{x} \in \mathbb{F}_2^n}[f(\mathbf{x}) \neq g(\mathbf{x})]$. Using $\mathsf{Var}(f_{\boldsymbol{\alpha}}) = \mathbb{E}(f_{\boldsymbol{\alpha}})(1 - \mathbb{E}(f_{\boldsymbol{\alpha}}))$, it is easy to check that $\min\left(\mathbb{E}(f_{\boldsymbol{\alpha}}), 1 - \mathbb{E}(f_{\boldsymbol{\alpha}})\right) \leq 2\mathsf{Var}(f_{\boldsymbol{\alpha}})$. The result now follows by taking $T = \overline{S}$ in (5.17) and combining with (5.29). □

5.3.4 Cryptographic Properties

An n-variable Boolean function f is δ-close to an s-junta if there is an s-junta g such that $\Pr_{\mathbf{x}\in\mathbb{F}_2^n}[f(\mathbf{x}) \neq g(\mathbf{x})] \leq \delta$. From the point of view of cryptographic design, it is undesirable for f to be δ-close to an s-junta for δ close to 0 and s smaller than n. Since otherwise, g is a good approximation of f and a cryptanalyst may replace f by g which may help in attacking a cipher which uses f as a building block. For example, in linear cryptanalysis the goal is to obtain g to be a linear function on a few variables such that it is a good approximation of f. To defend against such attacks, one usually requires f to not have any good linear approximation on a small number of variables. In particular, an m-resilient function cannot be approximated with probability different from $1/2$ by any linear function on m or smaller number of variables. A characterisation of resilient functions in terms of influence is given by Theorem 15 which shows that an n-variable function is m-resilient if and only if $(n-m)$-$\mathsf{inf}(f)$ takes its maximum value of 1.

The next result provides a characterisation of bent functions in terms of influence.

Theorem 19 *Let f be an n-variable Boolean function. Then f is bent if and only if for any non-empty $T \subseteq [n]$, $\mathsf{inf}_f(T) = 1 - 2^{\#T}$.*

Proof: First suppose that f is bent. So $W_f(\boldsymbol{\alpha}) = \pm 2^{-n/2}$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. From (2.14), it follows that $C_f(\mathbf{x}) = 0$ for all $\mathbf{0}_n \neq \mathbf{x} \in \mathbb{F}_2^n$. Consequently, from (5.6) we have that for any non-empty $T \subseteq [n]$, $\mathsf{inf}_f(T) = 1 - 2^{\#T}$.

Next we prove the converse. From (5.6), it follows that $\mathsf{inf}_f(T) = 1 - 2^{\#T}$ if and only if

$$\sum_{\mathbf{0}_n \neq \boldsymbol{\alpha} \leq \chi_T} C_f(\boldsymbol{\alpha}) \;=\; 0. \tag{5.30}$$

For $0 \leq i \leq 2^n - 1$, let $\mathsf{bin}_n(i)$ denote the n-bit binary representation of i. Let $\mathbf{M}$ be the $(2^n-1) \times (2^n-1)$ matrix whose rows and columns are indexed by the integers in $[2^n-1]$ such that the (i,j)-th entry of $\mathbf{M}$ is 1 if $\mathsf{bin}_n(j) \leq \mathsf{bin}_n(i)$ and otherwise the entry is 0. It is easy to verify that $\mathbf{M}$ is a lower triangular matrix whose diagonal elements are all 1. In particular, $\mathbf{M}$ is invertible.

Let $\mathbf{C} = [C_f(\mathsf{bin}_n(i))]_{i\in[2^n-1]}$ be the vector of auto-correlations of f at all the non-zero points in $\mathbb{F}_2^n$. The set of relations of the form (5.30) for all non-empty $T \subseteq [n]$ can be expressed as $\mathbf{M}\mathbf{C}^\top = \mathbf{0}^\top$. Since $\mathbf{M}$ is invertible, it follows that $\mathbf{C} = \mathbf{0}$, i.e. $C_f(\boldsymbol{\alpha}) = 0$ for all non-zero $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. From (2.13), it now follows that $W_f(\boldsymbol{\beta}) = \pm 2^{-n/2}$ for all $\boldsymbol{\beta} \in \mathbb{F}_2^n$ which

shows that f is bent. □

For functions satisfying propagation characteristics, somewhat less can be said. From (5.6), it follows that if f satisfies PC(k) then for any subset $\emptyset \neq T \subseteq [n]$ with $\#T = t \leq k$, $\mathsf{inf}_f(T) = 1 - 2^{-t}$ and so t-$\mathsf{inf}(f) = 1 - 2^{-t}$.

5.3.5 The Fourier Entropy/Influence Conjecture

The Fourier entropy $H(f)$ of f is defined to be the entropy of the probability distribution $\{W_f^2(\boldsymbol{\alpha})\}$ and is equal to

$$H(f) \;=\; -\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} W_f^2(\boldsymbol{\alpha}) \log W_f^2(\boldsymbol{\alpha}), \tag{5.31}$$

where log denotes $\log_2$ and the expressions $0 \log 0$ and $0 \log \frac{1}{0}$ are to be interpreted as 0. For $t \in [n]$, let

$$\rho_t(f) \;=\; \frac{H(f)/n}{t\text{-}\mathsf{inf}(f)}. \tag{5.32}$$

The Fourier entropy/influence conjecture [66] states that there is a universal constant C, such that for all Boolean functions f, $\rho_1(f) \leq C$. A general form of this conjecture is that there is a universal constant C_t, such that for all Boolean functions f and $t \in [1, n]$, $\rho_t(f) \leq C_t$. Since t-$\mathsf{inf}(f)$ increases monotonically with t, it follows that $\rho_t(f)$ decreases monotonically with t. So if the FEI conjecture holds, then the conjecture on $\rho_t(f)$ also holds for $t \geq 1$. The converse, i.e if the conjecture holds for some ρ_t with $t > 1$ then it also holds for ρ_1, need not be true.

Remark 8 *A weaker variant of the FEI conjecture replaces $H(f)$ by the min-entropy of the distribution $P_{\widehat{f}}(\omega)$. In a similar vein, one may consider the conjecture on $\rho_t(f)$ to be a weaker variant of the FEI conjecture.*

5.4 Pseudo-Influence

In this section, we define a quantity based on the auto-correlation function which we call the pseudo-influence of a Boolean function. The main reason for considering this notion is that it turns out to be the same as the notion of influence $J_f(T)$ introduced in [164]. We

make a thorough study of the basic properties of pseudo-influence. A consequence of this study is that pseudo-influence does not satisfy some of the basic desiderata that a notion of influence may be expected to satisfy, which is why we call it pseudo-influence. This shows that even though the quantity was termed 'influence' in [164], it is not a satisfactory notion of influence.

Suppose $f(\mathbf{X})$ is an n-variable Boolean function where $\mathbf{X} = (X_1, \ldots, X_n)$ and $\emptyset \neq T = \{i_1, \ldots, i_t\} \subseteq [n]$. We define pseudo-influence $\mathsf{PI}_f(T)$ of the set of variables $\{X_{i_1}, \ldots, X_{i_t}\}$ indexed by T on f in the following manner.

$$\mathsf{PI}_f(T) \;=\; \frac{1}{2^{\#T}} \left(\sum_{\boldsymbol{\alpha} \leq \chi_T} (-1)^{\mathsf{wt}(\alpha)} C_f(\boldsymbol{\alpha}) \right). \tag{5.33}$$

For a singleton set $T = \{i\}$, $\mathsf{PI}_f(T) = \mathsf{inf}_f(T) = \mathsf{inf}_f(i)$.

Let f be an n-variable function and t be an integer with $1 \leq t \leq n$. Then the t-pseudo-influence of f is the total pseudo-influence (scaled by $\binom{n}{t}$) obtained by summing the pseudo-influence of every set of t variables on the function f, i.e.,

$$t\text{-}\mathsf{PI}(f) \;=\; \frac{\sum_{\{T \subseteq [n] : \#T = t\}} \mathsf{PI}_f(T)}{\binom{n}{t}}. \tag{5.34}$$

The characterisation of pseudo-influence in terms of the Walsh transform is given by the following result.

Theorem 20 *Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq [n]$. Then*

$$\mathsf{PI}_f(T) \;=\; \sum_{\mathbf{u} \geq \chi_T} \left(W_f(\mathbf{u})\right)^2. \tag{5.35}$$

Consequently, for an integer t with $1 \leq t \leq n$,

$$t\text{-}\mathsf{PI}(f) \;=\; \frac{1}{\binom{n}{t}} \sum_{k=t}^{n} \binom{k}{t} P_{\widehat{f}}(k) \tag{5.36}$$

Proof: Let $\#T = t$. Let $E = \{\boldsymbol{\beta} \in \mathbb{F}_2^n : \boldsymbol{\beta} \leq \chi_{\overline{T}}\}$. Then $\#E = 2^{n-t}$ and $E^{\perp} = \{\boldsymbol{\alpha} \in \mathbb{F}_2^n : \boldsymbol{\alpha} \leq \chi_T\}$. From (5.33) and putting $\mathbf{a} = \mathbf{1}_n$, $\mathbf{b} = \mathbf{0}_n$ and $\psi = C_f$ in (2.5) we obtain the

following:

$$\mathsf{PI}_f(T) \;=\; \frac{1}{2^t}\sum_{\boldsymbol{\alpha}\le\chi_T}(-1)^{\mathsf{wt}(\boldsymbol{\alpha})}C_f(\boldsymbol{\alpha}) = \frac{1}{2^t}\sum_{\boldsymbol{\alpha}\le\chi_T}(-1)^{\langle\mathbf{1}_n,\boldsymbol{\alpha}\rangle}C_f(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\beta}\in\mathbf{1}_n+E}\widehat{C_f}(\boldsymbol{\beta}) = \sum_{\boldsymbol{\beta}\ge\chi_T}\widehat{C_f}(\boldsymbol{\beta}).$$

The result now follows from (2.13).

The expression for t-$\mathsf{PI}(f)$ can be seen as follows.

$$\begin{aligned} t\text{-}\mathsf{PI}(f) &= \frac{1}{\binom{n}{t}}\sum_{k=t}^{n}\sum_{\{\mathbf{u}:\mathsf{wt}(\mathbf{u})=k\}}\binom{k}{t}\left(W_f(\mathbf{u})\right)^2 \\ &= \frac{1}{\binom{n}{t}}\sum_{k=t}^{n}\binom{k}{t}\sum_{\{\mathbf{u}:\mathsf{wt}(\mathbf{u})=k\}}\left(W_f(\mathbf{u})\right)^2 \\ &= \frac{1}{\binom{n}{t}}\sum_{k=t}^{n}\binom{k}{t}P_{\widehat{f}}(k) \\ &= \frac{1}{\binom{n}{t}}\sum_{k=t}^{n}\binom{k}{t}P_{\widehat{f}}(k). \end{aligned} \tag{5.37}$$

□

The following result states the basic properties of the pseudo-influence.

Theorem 21 *Let f be an n-variable Boolean function and $\emptyset \neq T \subseteq S \subseteq [n]$. Then*

1. $0 \le \mathsf{PI}_f(T) \le 1$.
2. *If the function f is degenerate on the variables indexed by T, then* $\mathsf{PI}_f(T) = 0$.
3. $\mathsf{PI}_f(S) \le \mathsf{PI}_f(T)$.

Proof: The first point follows from Theorem 20 and Parseval's theorem. The third point also follows from Theorem 10.

Consider the second point. Suppose π is any permutation of $[n]$ and define $g(\mathbf{X})$ to be the function $f(X_{\pi(1)},\ldots,X_{\pi(n)})$. Then f is degenerate on the variables indexed by a set $U = \{i_1,\ldots,i_t\}$ if and only if g is degenerate on the variables indexed by the set $V = \{\pi(i_1),\ldots,\pi(i_t)\}$. Also, $\mathsf{inf}_f(U) = \mathsf{inf}_g(V)$. In view of this, we consider the set T to be $\{1,\ldots,t\}$.

For $\boldsymbol{\alpha}\in\mathbb{F}_2^t$ and $\mathbf{Y} = (X_{t+1},\ldots,X_n)$, let $f_{\boldsymbol{\alpha}}(\mathbf{Y}) = f(\boldsymbol{\alpha},\mathbf{Y})$. The function f is degenerate on the variables indexed by T if and only if $f_{\boldsymbol{\alpha}}(\mathbf{Y}) = f_{\boldsymbol{\beta}}(\mathbf{Y})$ for any $\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{F}_2^t$. We show

that the latter condition is equivalent to $f(\mathbf{X}) = f(\mathbf{X} \oplus \boldsymbol{\gamma})$ for any $\boldsymbol{\gamma} \leq \chi_T$. Note that by the choice of T, we have that for $\boldsymbol{\gamma} \leq \chi_T$, $\boldsymbol{\gamma} = (\boldsymbol{\delta}, \mathbf{0})$ for some $\boldsymbol{\delta} \in \mathbb{F}_2^t$. So it is sufficient to show that $f(\boldsymbol{\alpha}, \mathbf{Y}) = f((\boldsymbol{\alpha}, \mathbf{Y}) \oplus (\boldsymbol{\delta}, \mathbf{0}))$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^t$. The latter condition is equivalent to $f_{\boldsymbol{\alpha}}(\mathbf{Y}) = f_{\boldsymbol{\alpha} \oplus \boldsymbol{\delta}}(\mathbf{Y}) = f_{\boldsymbol{\beta}}(\mathbf{Y})$ where $\boldsymbol{\beta} = \boldsymbol{\alpha} \oplus \boldsymbol{\delta}$. This completes the proof that f is degenerate on the variables indexed by T if and only if $f(\mathbf{X}) = f(\mathbf{X} \oplus \boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \leq \chi_T$.

The condition $f(\mathbf{X}) = f(\mathbf{X} \oplus \boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \leq \chi_T$ is equivalent to $C_f(\boldsymbol{\gamma}) = 1$ for all $\boldsymbol{\gamma} \leq \chi_T$. So f is degenerate on the set of variables indexed by T if and only if $C_f(\boldsymbol{\gamma}) = 1$ for all $\boldsymbol{\gamma} \leq \chi_T$. Using this in the definition of pseudo-influence given by (5.33), we obtain the the second point. □

Theorem 21 states that if f is degenerate on the variables indexed by T, then $\mathsf{PI}_f(T) = 0$. The converse, however, is not true. Suppose f is an n-variable function such that $W_f(\mathbf{1}_n) = 0$ and let $T = [n]$. Then from (5.35), $\mathsf{PI}_f(T) = 0$. This example can be generalised. Suppose g is an n-variable, m-resilient function and let $f(\mathbf{X}) = \langle \mathbf{1}, \mathbf{X} \rangle \oplus g(\mathbf{X})$. Using (2.6), we have $W_f(\boldsymbol{\alpha}) = W_g(\mathbf{1} \oplus \boldsymbol{\alpha})$ for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n$. Since, g is m-resilient, $W_g(\boldsymbol{\omega}) = 0$ for all $\boldsymbol{\omega}$ with $\mathsf{wt}(\boldsymbol{\omega}) \leq m$. So $W_f(\boldsymbol{\alpha}) = 0$ for all $\boldsymbol{\alpha}$ with $\mathsf{wt}(\boldsymbol{\alpha}) \geq n - m$. Consequently, for any $\emptyset \neq T \subseteq [n]$, with $\#T \geq n - m$, it follows that $\mathsf{PI}_f(T) = 0$. There are known examples of non-degenerate resilient functions. See for example [150].

Remark 9 *By the above discussion,* $\mathsf{PI}_f(T)$ *can be zero even if* f *is non-degenerate on the variables indexed by* T*. Further, the third point of Theorem 21 shows that* $\mathsf{PI}_f(T)$ *is non-increasing with* T*. As a consequence, sub-additivity does not hold for* $\mathsf{PI}_f(T)$*. So* $\mathsf{PI}_f(T)$ *violates some of the basic desiderata that one may expect a notion of influence to fulfill.*

For $\mathbf{u} \in \mathbb{F}_2^n$ and $\emptyset \neq T \subseteq [n]$, $\mathbf{u} \geq \chi_T$ is equivalent to $\mathsf{supp}(\mathbf{u}) \supseteq T$ which in particular implies that $\mathsf{supp}(u) \cap T \neq \emptyset$. So from (5.8) and (5.35), we have the following result which states that influence is always at least as large as the pseudo-influence.

Proposition 5 *Let* f *be an* n*-variable Boolean function and* $\emptyset \neq T \subseteq [n]$*. Then* $\mathsf{inf}_f(T) \geq \mathsf{PI}_f(T)$*. Consequently,* t-$\mathsf{inf}(f) \geq t$-$\mathsf{PI}(f)$ *for* $1 \leq t \leq n$*.*

Theorem 22 *Let* $f(\mathbf{X})$ *be an* n*-variable Boolean function where* $\mathbf{X} = (X_1, \ldots, X_n)$ *and* t *be an integer with* $1 \leq t \leq n$*.*

1. t-$\mathsf{PI}(f)$ *takes its maximum value of* 1 *if and only if* f *is of the form* $f(\mathbf{X}) = \langle \mathbf{1}, \mathbf{X} \rangle$*.*
2. t-$\mathsf{PI}(f)$ *takes its minimum value of* 0 *if and only if* f *is of the form* $f(\mathbf{X}) = \langle \mathbf{1}, \mathbf{X} \rangle \oplus g(\mathbf{X})$*, where* $g(\mathbf{X})$ *is* $(n - t)$*-resilient.*

Proof: From (5.36), t-PI(f) takes its maximum value of 1 if and only if

$$\sum_{k=t}^{n} \binom{k}{t} P_{\widehat{f}}(k) \;=\; \binom{n}{t}. \tag{5.38}$$

If $f(\mathbf{X}) = \langle \mathbf{1}, \mathbf{X} \rangle$, then $P_{\widehat{f}}(n) = 1$ and $P_{\widehat{f}}(k) = 0$ for $0 \leq k \leq n-1$. On the other hand, if $f(\mathbf{X}) \neq \langle \mathbf{1}, \mathbf{X} \rangle$, then $P_{\widehat{f}}(n) < 1$ and we have

$$\begin{aligned}
&\binom{t}{t} P_{\widehat{f}}(t) + \binom{t+1}{t} P_{\widehat{f}}(t+1) + \cdots + \binom{n}{t} P_{\widehat{f}}(n) \\
&\leq \binom{n-1}{t} (P_{\widehat{f}}(0) + \cdots + P_{\widehat{f}}(n-1)) + \binom{n}{t} P_{\widehat{f}}(n) \\
&= \binom{n-1}{t} (1 - P_{\widehat{f}}(n)) + \binom{n}{t} P_{\widehat{f}}(n) < \binom{n}{t}.
\end{aligned}$$

This completes the proof of the first point.

For the second point, from (5.36), one may note that the values $P_{\widehat{f}}(0), \ldots, P_{\widehat{f}}(t-1)$ do not affect the expression for t-PI(f). So t-PI$(f) = 0$ if and only if $P_{\widehat{f}}(t) = \cdots = P_{\widehat{f}}(n) = 0$. The latter condition holds if and only if f is of the stated form. □

Using the second point of Theorem 22, it is possible to obtain examples of non-degenerate functions f such that t-PI(f) is 0.

Remark 10 *The quantity $J_f(T)$ (see* (5.4)*) was put forward by Tal [164] as a measure of influence of the set of variables indexed by T on the function f. It was shown in [164] that $J_f(T)$ is equal to the right hand side of* (5.35)*. So it follows that $J_f(T) = \mathsf{PI}_f(T)$. This is somewhat surprising since the definition of $J_f(T)$ given in* (5.4) *and that of $\mathsf{PI}_f(T)$ given in* (5.33) *are very different. It is perhaps only through the characterisations of both these quantities in terms of the Walsh transform that they can be seen to be equal. The quantity $\sum_{\{T:\#T=t\}} J_f(T)$ was considered in [164] and the expression* (5.36) *was also obtained in [164]. Since $J_f(T) = \mathsf{PI}_f(T)$, from Remark 9 it follows that $J_f(T)$ is not a satisfactory notion of influence.*

For an n-variable Boolean function f, define $L_{1,t} = \sum_{\mathbf{u}=t} |W_f(\mathbf{u})|$ and $W^{\geq t}(f) = \sum_{i \geq t} P_{\widehat{f}}(i)$. Lemma 31 of [164] showed that if for all t, t-PI$(f) \leq C \cdot \ell^t$ for some constant C, then $W^{\geq k}(f) \leq C \cdot e \cdot \ell \cdot e^{-(k-1)/(e\ell)}$ for all k. Lemma 34 of [164] showed that $L_{1,t}(f) \leq 2^t \cdot t$-PI$(f)$. Since Proposition 5 shows that t-INF$(f) \geq t$-PI(f) for $1 \leq t \leq n$, we obtain simple extensions of the Lemmas 31 and 34 of [164] by replacing t-PI(f) with t-INF(f) in the above statements.

Lemma 29 of [164] provides a converse of Lemma 31. This converse does not necessarily hold if t-$\mathsf{PI}(f)$ is replaced with t-$\mathsf{inf}(f)$. Lemmas 29 and 31 of [164] relate spectral tail bounds to bounds on pseudo-influence. We note that a spectral concentration result for t-$\mathsf{inf}(f)$ is given by Theorem 12.

5.5 Ben-Or and Linial Definition of Influence

The first notion of influence of a set of variables on a Boolean function was proposed by Ben-Or and Linial in [12]. In this section, we introduce this notion, prove some of its basic properties and show its relationship with the notion of influence defined in Section 5.3.

For an n-variable function f and $\emptyset \neq T \subseteq [n]$, with $t = \#T$, the notion of influence introduced in [12] is $\mathcal{I}_f(T)$ and is given by (5.1). For $t \in [n]$, we define

$$t\text{-}\mathcal{I}(f) \;=\; \frac{\sum_{\{T\subseteq[n]:\#T=t\}} \mathcal{I}_f(T)}{\binom{n}{t}}. \tag{5.39}$$

The following result provides an alternative description of $\mathcal{I}_f(T)$.

Proposition 6 *For an n-variable function f and $\emptyset \neq T \subseteq [n]$, with $t = \#T$,*

$$\mathcal{I}_f(T) \;=\; 1 - \frac{\#\left\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 = 1\right\}}{2^{n-t}} \tag{5.40}$$

$$\;=\; \frac{\#\left\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : \left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 \neq 1\right\}}{2^{n-t}}, \tag{5.41}$$

where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}}\leftarrow\boldsymbol{\alpha}}$.

Proof: From (5.1), it clearly follows that

$$\begin{aligned}
\mathcal{I}_f(T) &= 1 - \frac{\#\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : f_{\boldsymbol{\alpha}} \text{ is constant}\}}{2^{n-t}} \\
&= 1 - \frac{\#\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : \mathsf{wt}(f_{\boldsymbol{\alpha}}) = 0,\ \text{or } 2^t\}}{2^{n-t}} \\
&= 1 - \frac{\#\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t) = \pm 1\}}{2^{n-t}}.
\end{aligned}$$

This shows (5.40), and (5.41) follows directly from (5.40). □

Some basic properties of $\mathcal{I}_f(T)$ are as follows.

Theorem 23 *Let f be an n-variable function and $\emptyset \neq T \subseteq S \subseteq [n]$. Let $\#T = t$.*

1. $0 \leq \mathcal{I}_f(T) \leq 1$.
2. $\mathcal{I}_f(T) = 0$ *if and only if f is degenerate on the variables indexed by T.*
3. $\mathcal{I}_f(T) = 1$ *if and only if $f_{\boldsymbol{\alpha}}$ is a non-constant function for every $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}$. In particular, if $T = [n]$, then $\mathcal{I}_f(T) = 1$.*
4. $\mathcal{I}_f(T) \leq \mathcal{I}_f(S)$.

Proof: The first point is obvious.

For the second point, using (5.40) note that $\mathcal{I}_f(T) = 0$ if and only if for every $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, $W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t) = \pm 1$, i.e., if and only if $\mathsf{wt}(f_{\boldsymbol{\alpha}}) = 0,\ \text{or } 2^t$, i.e., if and only if $f_{\boldsymbol{\alpha}}$ is constant. The latter condition holds if and only if the variables indexed by T have no effect on the value of f, i.e., if and only if f is degenerate on the variables indexed by T.

To see the third point, note that $\mathcal{I}_f(T) = 1$ if and only if for every $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, $\left(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t)\right)^2 \neq 1$, which holds if and only if $f_{\boldsymbol{\alpha}}$ is a non-constant function.

Let $\#S = s$. For the fourth point, it is sufficient to consider $s = t+1$, since otherwise, we may define a sequence of sets $T \subset S_1 \subset S_2 \subset \cdots \subset S$, with $\#T + 1 = \#S_1$, $\#S_1 + 1 = \#S_2$, $\ldots$, and argue $\mathcal{I}_f(T) \leq \mathcal{I}_f(S_1) \leq \cdots \leq \mathcal{I}_f(S)$. Further, without loss of generality, we assume $T = \{n-t+1, \ldots, n\}$ and $S = \{n-t, \ldots, n\}$ as otherwise, we may apply an appropriate permutation on the variables to ensure this condition. Then $\overline{T} = \{1, \ldots, n-t\}$ and $\overline{S} = \{1, \ldots, n-t-1\}$.

Let $\mathcal{T} = \{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : f_{\boldsymbol{\alpha}} \text{ is constant}\}$ and $\mathcal{S} = \{\boldsymbol{\beta} \in \mathbb{F}_2^{n-t-1} : f_{\boldsymbol{\beta}} \text{ is constant}\}$, where $f_{\boldsymbol{\beta}}$ is a shorthand for $f_{\mathbf{X}_{\overline{S}} \leftarrow \boldsymbol{\beta}}$. Note that if $\boldsymbol{\beta} \in \mathcal{S}$, then $(\boldsymbol{\beta}, 0), (\boldsymbol{\beta}, 1) \in \mathcal{T}$. So $\#\mathcal{T} \geq 2\#\mathcal{S}$ which implies

$$\frac{\#\mathcal{T}}{2^{n-t}} \geq \frac{2\#\mathcal{S}}{2^{n-t}} \geq \frac{\#\mathcal{S}}{2^{n-t-1}}.$$

Consequently,

$$\mathcal{I}_f(T) = 1 - \frac{\#\mathcal{T}}{2^{n-t}} \leq 1 - \frac{\#\mathcal{S}}{2^{n-t-1}} = \mathcal{I}_f(S).$$

□

Remark 11 *We note that the sub-additivity property does not hold for $\mathcal{I}_f(T)$. As an example, consider a 6-variable function f which maps $\mathbf{0}_6$ to 1 and all other elements of $\mathbb{F}_2^6$ to 0; let $S = \{4, 5, 6\}$ and $T = \{2, 3, 6\}$. Then $\mathcal{I}_f(S \cup T) = 1/2 > 1/8 + 1/8 = \mathcal{I}_f(S) + \mathcal{I}_f(T)$.*

Next, we show that the Ben-Or and Linial notion of influence is always at least as much as the notion of influence defined in (5.6).

Theorem 24 *Let f be an n-variable function and $\emptyset \neq T \subseteq [n]$. Then $\mathsf{inf}_f(T) \leq \mathcal{I}_f(T)$. Further, equality holds if and only if $(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2 = 0$ or 1 for each $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, where $f_{\boldsymbol{\alpha}}$ denotes $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}$.*

Proof: We rewrite (5.16) in the following form.

$$\mathsf{inf}_f(T) = \frac{1}{2^{n-t}} \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(1 - (W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2\right). \tag{5.42}$$

Consider the expressions for $\mathsf{inf}_f(T)$ and $\mathcal{I}_f(T)$ given by (5.42) and (5.41) respectively. Both the expressions are sums over $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$. Suppose $\boldsymbol{\alpha}$ is such that $(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2 = 1$. The contribution of such an $\boldsymbol{\alpha}$ to both (5.42) and (5.41) is 0. Next suppose $(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2 \neq 1$; the contribution of such an $\boldsymbol{\alpha}$ to (5.41) is 1 and the contribution to (5.42) is at most 1, and the value 1 is achieved if and only if $W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t) = 0$. □

One may compare the properties of $\mathcal{I}_f(T)$ given by Theorem 23 to the desiderata that a notion of influence may be expected to satisfy (see the discussion before Theorem 14). The measure $\mathcal{I}_f(T)$ satisfies some of the desiderata, namely, it is between 0 and 1; takes the value 0 if and only if f is degenerate on the variables indexed by T; and it is monotone increasing with the size of T. On the other hand, as noted above, it does not satisfy the sub-additivity property.

Compared to $\mathsf{inf}_f(T)$, the value of $\mathcal{I}_f(T)$ rises quite sharply. To see this, it is useful to view the following expressions for the two quantities.

$$2^{n-t} \times \mathsf{inf}_f(T) = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}} \left(1 - (W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2\right), \tag{5.43}$$

$$2^{n-t} \times \mathcal{I}_f(T) = \#\left\{\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t} : (W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2 \neq 1\right\}. \tag{5.44}$$

Suppose $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$ is such that $f_{\boldsymbol{\alpha}}$ is a non-constant function, so that $(W_{f_{\boldsymbol{\alpha}}}(\mathbf{0}_t))^2 \neq 1$. Then such an $\boldsymbol{\alpha}$ contributes 1 to (5.44), while it contributes a value which is at most 1 to (5.43).

More generally, $\boldsymbol{\alpha}$ contributes either 0 or 1 to (5.44) according as $f_{\boldsymbol{\alpha}}$ is constant or non-constant; on the other hand, the contribution of $\boldsymbol{\alpha}$ to (5.43) is more granular. Consequently, the value of $\mathcal{I}_f(T)$ rises more sharply than the value of $\mathsf{inf}_f(T)$. In particular, if f and g are two distinct functions such that for all $\boldsymbol{\alpha}$, both $f_{\boldsymbol{\alpha}}$ and $g_{\boldsymbol{\alpha}}$ are non-constant functions, then both $\mathcal{I}_f(T)$ and $\mathcal{I}_g(T)$ will be necessarily be equal to 1, whereas the values of $\mathsf{inf}_f(T)$ and $\mathsf{inf}_g(T)$ are neither necessarily 1 nor necessarily equal. In other words, the discerning power of $\mathcal{I}_f(T)$ as a measure of influence is less than that of $\mathsf{inf}_f(T)$, i.e., $\mathcal{I}_f(T)$ is a more coarse measure of influence. So while both $\mathsf{inf}_f(T)$ and $\mathcal{I}_f(T)$ share some intuitive basic properties expected of a definition of influence, the facts that $\mathcal{I}_f(T)$ does not satisfy sub-additivity and has less discerning power make it a less satisfactory measure of influence compared to $\mathsf{inf}_f(T)$.

Theorem 24 shows that $\mathsf{inf}_f(T) \leq \mathcal{I}_f(T)$. The difference between $\mathcal{I}_f(T)$ and $\mathsf{inf}_f(T)$ can be quite large. For example, if we take $f(\mathbf{X}) = X_1 \cdots X_n$ (i.e., the Boolean AND function), then $\mathcal{I}_f([n]) = 1$ while $\mathsf{inf}_f([n]) = 1 - (1 - 1/2^{n-1})^2$. In other words, the influence of the set of all variables as measured by $\mathcal{I}_f$ is 1, while the influence as measured by inf_f is close to 0. The influence of $[n]$ on the degenerate n-variable constant all-zero function is 0 as measured by both $\mathcal{I}_f$ and inf_f. The AND function differs from the all-zero function by a single bit and so one would expect the influence of $[n]$ to remain close to 0. This is indeed the case for inf_f, while for $\mathcal{I}_f$ the value jumps to 1. The example of the AND function can be generalised to a balanced function in the following manner. Let $1 \leq t < n$ and define $f(X_1, X_2, \ldots, X_n) = X_1 \cdots X_t \oplus X_{t+1} \oplus \cdots \oplus X_n$. It is easy to verify that f is balanced. Let $T = \{1, \ldots, t\}$. One may check that $\mathcal{I}_f(T) = 1$ and $\mathsf{inf}_f(T) = 1 - (1 - 1/2^{t-1})^2$. As in the case of the AND function, it can be argued that one would expect the influence of T to be close to 0 rather than being equal to 1.

The following result characterises the minimum and maximum values of t-$\mathcal{I}(f)$.

Theorem 25 *Let f be an n-variable Boolean function and t be an integer with $1 \leq t \leq n$.*

1. *t-$\mathcal{I}(f)$ takes its maximum value of 1 if and only if for every subset T of $[n]$ of size t, and for every $\boldsymbol{\alpha} \in \mathbb{F}_2^{n-t}$, the function $f_{\mathbf{X}_{\overline{T}} \leftarrow \boldsymbol{\alpha}}(\mathbf{X}_T)$ is non-constant.*

2. *t-$\mathcal{I}(f)$ takes its minimum value of 0 if and only if f is a constant function.*

Proof: The proof of the first point follows from the third point of Theorem 23.

For the second point, we note that if f is a constant function, then from (5.1), $\mathcal{I}_f(T) = 0$ for every subset T of $[n]$ and so t-$\mathcal{I}(f)$. On the other hand, if t-$\mathcal{I}(f) = 0$, then from

Theorem 24, it follows that $t\text{-}\mathsf{inf}(f) = 0$ and so from the second point of Theorem 15 we have that f is a constant function. □

Remark 12 *Upper bounds on $\mathcal{I}_f(T)$ for T with bounded size have been proved in [2]. Since $\mathsf{inf}_f(T) \leq \mathcal{I}_f(T)$, it follows that these upper bounds also hold for $\mathsf{inf}_f(T)$.*

5.6 Discussion

We have introduced a new definition of influence of a set of variables on a Boolean function which is based on the auto-correlation function. Using the new definition, we have proved a number of results. In this section, we highlight the new insights into Boolean functions that are obtained from the new results which follow from the new definition.

As proved in Section 5.3.2, the quantity $I_f(T)$ defined in [62, 23] is half the value of the influence (namely, $\mathsf{inf}_f(T)$) that we have defined. Some results for $I_f(T)$ have been obtained earlier. Remark 7 mentions the results which were previously obtained in [62, 23]. The quantity $I_f(T)$ was used in [62, 23] as a tool for junta testing. The crucial result for such testing is Proposition 4. We have provided a new and simpler proof of this proposition. Apart from Proposition 4 and the results mentioned in Remark 7, all other results in Section 5.3 and its various subsections appear for the first time in this paper. We highlight interesting aspects of some of the new results, particularly those aspects which arise due to the auto-correlation function based definition.

Theorem 17 connects total influence to the path expansion property of a set of vertices A of the hypercube. This result provides a geometric interpretation of the notion of influence which generalises the well known connection of the total influence of a single variable to the edge expansion property of A. The geometric interpretation of total influence in terms of path expansion is obtained through the connection of the auto-correlation function to path expansion and the new definition of influence using the auto-correlation function. The Fourier/Walsh transform and the auto-correlation function are well studied tools in the theory of Boolean functions. In Proposition 2 and Remark 6 we have explained the new geometric insight into these tools that our results provide.

The notion of influence has been studied for a long time, but has been restricted mostly to issues in theoretical computer science. On the other hand, the notions of bent functions and resilient functions have also been studied for a long time in the coding theory and cryptography literature. Our results provide a previously unknown bridge between the notion

of influence on the one hand, and the notions of bent and resilient functions on the other hand. The first point of Theorem 15 provides a characterisation of resilient functions in terms of total influence. Theorem 19 provides a characterisation of bent functions in terms of influence. Theorem 15 is itself based on the characterisation of total influence in terms of Fourier/Walsh transform, while the proof of Theorem 19 uses the auto-correlation based definition of influence. These new results provide interesting new insights into the connection between aspects of Boolean functions studied in theoretical computer science and in coding theory and cryptography.

Remark 10 and the discussion following it mention the results on pseudo-influence which were previously obtained in [164]. The other results in Section 5.4 are new to this work. In particular, the inadequacy of pseudo-influence as a notion of influence is obtained as a consequence of Theorem 14, and the characterisation of the conditions under which the total pseudo-influence achieves its minimum and maximum values are given in Theorem 22.

All results in Section 5.5 on the BL definition of influence are new to this paper. These results establish the basic properties of this notion of influence. We provide a detailed comparison of the BL definition of influence and the auto-correlation function based definition of influence which highlight why the BL definition is less satisfactory than the auto-correlation function based definition as a measure of influence.

5.7 Conclusion

We introduced a definition of influence of a set of variables on a Boolean function using the auto-correlation function. The basic theory around the notion of influence has been carefully developed and several well known results on the influence of a single variable have been generalised. New characterisations of resilient and bent functions in terms of influence have been obtained. A previously introduced [62, 23] measure of influence of a set of variables is shown to be half the value of the influence that we introduce. We also defined a notion of pseudo-influence, argued that it is not a satisfactory measure of influence and showed that pseudo-influence is equal to a measure of influence previously defined in [164]. Finally, we studied in details the definition of influence given by Ben-Or and Linial [12] and brought out its relation to the auto-correlation based notion of influence.

Chapter 6

A lower bound on the constant in the Fourier min-entropy/influence conjecture

In mathematical research, determining bounds for constants in inequalities holds intrinsic mathematical value and is pivotal for advancing our understanding of problems. These bounds offer crucial insights into the structure of mathematical objects, guiding further investigations. One well-known example in mathematics where finding the exact value of the constant is of significant interest is the Berry-Esseen theorem [57, 16]. This theorem is fundamental in probability theory and statistics, estimating errors between the distribution of a sum of random variables and the normal distribution. Esseen showed that the value of the constant cannot be less than 0.4097 [79]. Mathematicians have long been interested in improving the upper bounds for the constants involved in this theorem to reduce the gap factor between the upper and lower bounds. Over time, there has been a gradual reduction in the upper bound of the Berry-Essen constant for the independent and identically distributed (iid) scenario. It started at the original value of 7.59 as calculated by Esseen in 1942 [57], and through subsequent research, it was significantly reduced to 0.7882 by van Beek in 1972 [170], further down to 0.7655 by Shiganov in 1986 [158], and continued to decrease with contributions from Shevtsova in 2007 (0.7056) [156] and 2008 (0.7005) [106], Tyurin in 2009 (0.5894) [167], and Korolev and Shevtsova in 2010 (0.5129) [107]. Another reduction was achieved by Tyurin in 2010 [168], bringing bound of the Berry-Essen constant to 0.4785. The best known bound by 2012, according to Shevtsova's work in 2011 [157], puts the constant at less than 0.4748 [1]. Likewise, researchers have actively explored scenarios involving independent but non-identically distributed variables. While Essen's lower bound for the iid case from 1956 remains applicable here, the upper bound varies from the iid case. In this scenario also, the upper bounds for the Berry-Essen constant have significantly decreased over time. The original estimate of 7.59 by Esseen in 1942 [57] has been lowered to 0.9051 by Zolotarev (1967) [181], 0.7975 by van Beek (1972) [170], 0.7915 by Shiganov (1986) [158], and further reduced to 0.6379 and 0.5606 by Tyurin in 2009 and 2010 respectively [167, 168]. As of 2011, the best estimate stands at 0.5600, obtained by Shevtsova [157].

In the context of the Analysis of Boolean functions, one long-standing open problem is

to prove the Fourier entropy/influence (FEI) conjecture, presented in Chapter 3. Naturally, this problem can be approached as an attempt to establish the bounds for the FEI constant. It's worth noting that determining the upper limit of the FEI constant essentially resolves the conjecture itself. Despite years of research, the FEI Conjecture remains unresolved to this day. Another potential research avenue involves exploring the lower bound of the FEI constant. There are two reasons for attempting to determine the lower bound of the FEI constant: firstly, if the conjecture is proven in the future, much like the Berry-Esseen constant, approaching to the precise value of the FEI constant from any direction will become of significant mathematical interest. Secondly, while pursuing this lower bound, there is a possibility that specific constructions might shed new light on the properties examined in potential counterexamples to the conjecture. There are works available in the literature attempting to find the lower bound of the constant in the Fourier entropy/influence (FEI) conjecture. Hence, it is evident that the pursuit of a lower bound on the constant in the Fourier min-entropy/influence conjecture, a specialized case of the FEI conjecture, possesses a similar inherent mathematical value. Furthermore, it promises to enhance our understanding of the structure of Boolean functions.

In this chapter we first describe a new construction of Boolean functions. A specific instance of our construction provides a 30-variable Boolean function having min-entropy/influence ratio to be $128/45 \approx 2.8444$ which is presently the highest known value of this ratio that is achieved by any Boolean function. Correspondingly, $128/45$ is also presently the best known lower bound on the universal constant of the Fourier min-entropy/influence conjecture.

6.1 Introduction

A longstanding open problem in the field of analysis of Boolean functions is the Fourier Entropy/Influence (FEI) conjecture made by Friedgut and Kalai in 1996 [66]. The FEI conjecture states that there is a universal constant C such that $H(f) \leq C \cdot \mathsf{inf}(f)$ for any Boolean function f, where $H(f)$ and $\mathsf{inf}(f)$ denote the Fourier entropy and the total influence of f respectively. For an explanation of the motivation behind the FEI conjecture, please refer Chapter 3.

The conjecture was verified for various families of Boolean functions (e.g., symmetric functions [131], read-once formulas [130, 42], decision trees of constant average depth [171], read-k decision trees for constant k [171], functions with exponentially small influence or with linear entropy [154], random linear threshold functions [41], cryptographic Boolean

functions [68], random functions [51]), but is still open for the class of all Boolean functions.

There has also been research in obtaining lower bounds on the constant C in the FEI conjecture. To show that C is at least some value δ it is sufficient to show the existence of a Boolean function whose entropy/influence ratio is δ. The first lower bound of 4.615 was obtained by O'Donnell et al. in [131]. Later O'Donnell and Tan [130] provided a recursive construction of Boolean functions which showed how to construct a function for which the value of the entropy/influence ratio is at least 6.278944 [86]. The presently best known lower bound on C is 6.454784. This bound was shown by Hod [86] using an extensive asymptotic analysis.

The Fourier min-entropy/influence (FMEI) conjecture was put forward by O'Donnell et al. in 2011 [131]. The FMEI conjecture states that there is a universal constant D such that $H_\infty(f) \leq D \cdot \mathsf{inf}(f)$ for any Boolean function f, where $H_\infty(f)$ is the Fourier min-entropy of f. The FMEI conjecture is weaker than the FEI conjecture in the sense that settling the FEI conjecture will also settle the FMEI conjecture, but the converse is not true. It was observed in [41, 131] that as a consequence of the Kahn-Kalai-Linial theorem [92] the FMEI conjecture holds for monotone functions and linear threshold functions. The FMEI conjecture for "regular" read-k DNFs was established by Shalev [154]. More recently, Arunachalam et al. [5] have shown that the FMEI holds for read-k DNF for constant k.

To the best of our knowledge, till date there has been no work on obtaining lower bounds on the universal constant of the FMEI conjecture. Since the FMEI conjecture is weaker than the FEI conjecture, any upper bound on the universal constant of the FEI conjecture is also an upper bound on the universal constant of the FMEI conjecture. This, however, does not hold for lower bounds, i.e. a lower bound on the universal constant of the FEI conjecture is not necessarily a lower bound on the universal conjecture of the FMEI conjecture.

6.1.1 Our results

The purpose of this work is to obtain a lower bound on the universal constant D of the FMEI conjecture. As in the case of the FEI conjecture, to show that D is at least δ, it is sufficient to show the existence of a Boolean function for which the min-entropy/influence ratio is δ. An exhaustive search over all n-variable Boolean functions, with $1 \leq n \leq 5$, shows that the maximum value of min-entropy/influence ratio that is achieved by functions of at most 5 variables is $16/7 \approx 2.285714$. Since an exhaustive search becomes infeasible for $n \geq 6$, it is required to obtain some method of constructing Boolean functions for which the

min-entropy/influence ratio is greater than 16/7.

We first considered the recursive construction of O'Donnell and Tan [130], since this construction proved to be useful for showing a lower bound on the constant of the FEI conjecture. To analyse this construction in the context of the FMEI conjecture, we derived an expression for the min-entropy of the functions obtained using this construction. Since the construction is recursive, one needs an initial function to start the recursion. We performed an exhaustive search over all possible 5-variable initial functions. This yielded a 25-variable function having min-entropy/influence ratio equal to $512/225 \approx 2.275556$. This unfortunately is not useful since 512/225 is less than 16/7, the maximum value of min-entropy/influence ratio that is obtained by exhaustive search over all 5-variable functions. The 25-variable function is obtained in the first step of the O'Donnell-Tan recursion. Considering further steps of the recursion does not result in a higher value of the min-entropy/influence ratio. We identified an alternative recursive construction of Boolean functions which provides a lower bound on the constant of the FEI conjecture which is equal to that obtained from the O'Donnell-Tan construction. This alternative construction, however, does not improve upon the lower bound on the constant of the FMEI conjecture that is obtained from the O'Donnell-Tan construction. Further, we did not find any way to apply the asymptotic constructions given by Hod [86] in the context of the FEI conjecture for obtaining lower bounds on the constant in the FMEI conjecture.

Our main result is a new construction of Boolean functions. In simple terms, the construction takes an n-variable function g and constructs an $(n+1)$-variable palindromic function g_0. An $n(n+1)$-variable function G_0 is then constructed by taking the 'disjoint composition' (see 6.2) of g_0 and g. Under certain conditions on g, the min-entropy/influence ratio of G_0 is greater than that of g. By searching over all appropriate 5-variable functions g, we obtain a 30-variable function G_0 having min-entropy/influence ratio to be equal to $128/45 \approx 2.844444$. In fact, we obtain a total of 384 such functions G_0. The value 128/45 is presently the highest achieved value of min-entropy/influence ratio and correspondingly is presently the best known lower bound on D.

In the final section, we provide a brief description of some experiments that we have carried out for symmetric and rotation-symmetric Boolean functions. Based on these experiments, we put forward a new conjecture on entropy/influence and the min-entropy/influence ratios of symmetric Boolean functions.

6.2 Background

We have already presented the FEI conjecture in Chapter 3. Here, we begin by stating the FMEI conjecture, which establishes a connection between min-entropy and influence.

The Fourier Min-entropy/influence (FMEI) conjecture [131]. There exists a universal constant D such that for any integer $n \geq 1$ and for any n-variable Boolean function f, $H_\infty(f) \leq D \cdot \mathsf{inf}(f)$.

Composition. For positive integers n and k, an (n,k) vectorial Boolean function (also called an S-box) is a map $\mathscr{G} : \mathbb{F}_2^n \to \mathbb{F}_2^k$. The function $\mathscr{G}$ can be written as $\mathscr{G}(\mathbf{X}) = (g_1(\mathbf{X}), \ldots, g_k(\mathbf{X}))$, where $g_1, \ldots, g_k$ are n-variable Boolean functions. Given a k-variable Boolean function f and an (n,k) vectorial Boolean function $\mathscr{G}$, their composition is the n-variable Boolean function $(f \circ \mathscr{G})(\mathbf{X}) = f(g_1(\mathbf{X}), \ldots, g_k(\mathbf{X}))$. The Walsh transform of $f \circ \mathscr{G}$ is given by the following result.

Theorem 26 *[78] Let $\mathscr{G}$ be an (n,k) vectorial Boolean function and f be a k-variable Boolean function. Then for any $\mathbf{u} \in \mathbb{F}_2^n$,*

$$W_{f\circ\mathscr{G}}(\mathbf{u}) = \textstyle\sum_{\mathbf{v}\in\mathbb{F}_2^k} W_f(\mathbf{v}) W_{(l_\mathbf{v}\circ\mathscr{G})}(\mathbf{u}), \tag{6.1}$$

where $(l_\mathbf{v} \circ \mathscr{G})(\mathbf{X}) = \langle \mathbf{v}, \mathscr{G}(\mathbf{X}) \rangle$.

Let k and l be positive integers and $n = kl$. For $\mathbf{x} \in \mathbb{F}_2^n$ and $1 \leq i \leq k$, by $\mathbf{x}^{(i)}$ we denote the vector $(x_{(i-1)l+1}, \ldots, x_{il}) \in \mathbb{F}_2^l$. By a slight abuse of notation, we will write $\mathbf{X} = (\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(k)})$. Let f and g be Boolean functions on k and l variables respectively and $n = kl$. Let $\mathscr{G}$ be the (n,k) vectorial Boolean function given by $\mathscr{G}(\mathbf{X}) = (g(\mathbf{X}^{(1)}), \ldots, g(\mathbf{X}^{(k)}))$. The *disjoint composition* of f and g, which we will denote as $f \diamond g$, is the n-variable Boolean function $f \circ \mathscr{G}$, i.e.

$$(f \diamond g)(\mathbf{X}) = (f \circ \mathscr{G})(\mathbf{X}) = f(g(\mathbf{X}^{(1)}), \ldots, g(\mathbf{X}^{(k)})). \tag{6.2}$$

The following result provides the entropy and influence of $f \diamond g$.

Theorem 27 (simplified form of Proposition 2 in [130]) *Let f and g be two Boolean functions. Then,*

1. $\mathsf{inf}(f \diamond g) = \mathsf{inf}(g) \cdot \mathsf{inf}(f)$.

2. *If g is balanced, then* $H(f \diamond g) = H(f) + H(g) \cdot \mathsf{inf}(f)$.

O'Donnell-Tan recursive construction. The following recursive construction of Boolean functions was introduced by O'Donnell and Tan [130]. Let g be an l-variable Boolean function. Using g, a sequence of Boolean functions f_m, $m \geq 0$, is defined in the following manner.

$$\left.\begin{array}{rcl} f_0 & = & g, \\ f_m & = & g \diamond f_{m-1} \quad \text{if } m \geq 1. \end{array}\right\} \tag{6.3}$$

It is easy to see that for $m \geq 0$, f_m is a map from $\mathbb{F}_2^{l^{m+1}} \to \mathbb{F}_2$. For the recursion defined in (6.3), in the case where the initial function g is balanced, the following was proved in [130].

$$\frac{H(f_m)}{\mathsf{inf}(f_m)} = \frac{H(g)}{\mathsf{inf}(g)} + \frac{H(g)}{\mathsf{inf}(g)(\mathsf{inf}(g)-1)} - \frac{H(g)}{\mathsf{inf}(g)^{m+1}(\mathsf{inf}(g)-1)}. \tag{6.4}$$

Consequently, $\lim_{m\to\infty} H(f_m)/\mathsf{inf}(f_m) = H(g)/(\mathsf{inf}(g)-1)$. So for any Boolean function g, $H(g)/(\mathsf{inf}(g)-1)$ is a lower bound on the constant in the FEI conjecture.

6.3 Min-Entropy of disjoint composition

We wish to compute the min-entropy of disjoint composition. We start with the following result which is somewhat more general than what we need.

Theorem 28 *Let k and l be positive integers and $n = kl$. Let $\mathscr{G}$ be an (n,k) vectorial Boolean function such that $\mathscr{G}(\mathbf{X}) = (g_1(\mathbf{X}^{(1)}), \ldots, g_k(\mathbf{X}^{(k)}))$, where $g_1, \ldots, g_k$ are l-variable balanced Boolean functions. Then for any k-variable Boolean function f,*

$$W_{f\circ\mathscr{G}}(\mathbf{u}) = \begin{cases} W_f(\mathbf{0}_k) & \text{if } \mathbf{u} = \mathbf{0}_n, \\ W_f(\mathbf{w_u}) \prod_{i \in \mathsf{supp}(\mathbf{w_u})} W_{g_i}\left(\mathbf{u}^{(i)}\right) & \text{otherwise.} \end{cases} \tag{6.5}$$

In (6.5), for $\mathbf{u} \in \mathbb{F}_2^n$ written as $\mathbf{u} = (\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)})$, by $\mathbf{w_u}$ we denote the vector in $\mathbb{F}_2^k$ whose i-th position, $1 \leq i \leq k$, is 1 if and only if $\mathbf{u}^{(i)} \neq \mathbf{0}_l$, i.e. $\mathbf{w_u}$ encodes whether the -bit blocks of $\mathbf{u}$ are zero or not.

Proof: The proof follows from an application of Theorem 26.

Note that for $\mathbf{v} = (v_1, \ldots, v_k) \in \mathbb{F}_2^k$, $(l_{\mathbf{v}} \circ \mathscr{G})(\mathbf{X}) = v_1 \cdot g_1(\mathbf{X}^{(1)}) \oplus \cdots \oplus v_k \cdot g_k(\mathbf{X}^{(k)})$. So for $\mathbf{u} = (\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)}) \in \mathbb{F}_2^n$,

$$\begin{aligned} W_{(l_{\mathbf{v}} \circ \mathscr{G})}(\mathbf{u}) &= \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{(l_{\mathbf{v}} \circ \mathscr{G})(\mathbf{x}) \oplus \langle \mathbf{u}, \mathbf{x} \rangle} \\ &= \frac{1}{2^n} \sum_{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)} \in \mathbb{F}_2^l} (-1)^{v_1 \cdot g_1\left(\mathbf{x}^{(1)}\right) \oplus \langle \mathbf{u}^{(1)}, \mathbf{x}^{(1)} \rangle \oplus \cdots \oplus v_k \cdot g_k\left(\mathbf{x}^{(k)}\right) \oplus \langle \mathbf{u}^{(k)}, \mathbf{x}^{(k)} \rangle} \\ &= \prod_{i \in [k]} \frac{1}{2^l} \sum_{\mathbf{x}^{(i)} \in \mathbb{F}_2^l} (-1)^{v_i \cdot g_i\left(\mathbf{x}^{(i)}\right) \oplus \langle \mathbf{u}^{(i)}, \mathbf{x}^{(i)} \rangle}. \end{aligned}$$

For $i \in [k]$, let $B_i\left(v_i, \mathbf{u}^{(i)}\right) = \frac{1}{2^l} \sum_{\mathbf{x}^{(i)} \in \mathbb{F}_2^l} (-1)^{v_i \cdot g_i\left(\mathbf{x}^{(i)}\right) \oplus \langle \mathbf{u}^{(i)}, \mathbf{x}^{(i)} \rangle}$. Using (6.1) we have,

$$W_{f \circ \mathscr{G}}(\mathbf{u}) = \sum_{\mathbf{v} \in \mathbb{F}_2^k} W_f(\mathbf{v}) W_{(l_{\mathbf{v}} \circ \mathscr{G})}(\mathbf{u}) = \sum_{\mathbf{v} \in \mathbb{F}_2^k} W_f(\mathbf{v}) \prod_{i \in [k]} B_i\left(v_i, \mathbf{u}^{(i)}\right). \tag{6.6}$$

Let us now consider $B_i\left(v_i, \mathbf{u}^{(i)}\right)$. Note that $B_i\left(0, \mathbf{u}^{(i)}\right)$ is equal to 1 or 0 according as $\mathbf{u}^{(i)}$ is equal to $\mathbf{0}_l$ or not. Further, $B_i\left(1, \mathbf{u}^{(i)}\right) = W_{g_i}\left(\mathbf{u}^{(i)}\right)$. Since it is given that g_i is balanced, so $B_i\left(1, \mathbf{0}_l\right) = 0$.

For $\mathbf{u} \in \mathbb{F}_2^n$, the i-th bit of $\mathbf{w}_{\mathbf{u}}$ is 1 if and only if the i-th block of $\mathbf{u}$ is non-zero. For $\mathbf{v} \in \mathbf{F}_2^k$ such that $\mathbf{v} \neq \mathbf{w}_{\mathbf{u}}$, there is a $j \in [k]$ such that either $v_j = 0$ and $\mathbf{u}^{(j)} \neq \mathbf{0}_l$, or $v_j = 1$ and $\mathbf{u}^{(j)} = \mathbf{0}_l$; in either case, $B_j\left(v_j, \mathbf{u}^{(j)}\right) = 0$ and so $\prod_{i \in [k]} B_i\left(v_i, \mathbf{u}^{(i)}\right) = 0$. On the other hand, for $\mathbf{v} = \mathbf{w}_{\mathbf{u}}$, if $v_i = 0$ then $\mathbf{u}^{(i)} = \mathbf{0}_l$ which implies $B_i\left(v_i, \mathbf{u}^{(i)}\right) = 1$; and if $v_i = 1$ then $\mathbf{u}^{(i)} \neq \mathbf{0}_l$ which implies $B_i\left(v_i, \mathbf{u}^{(i)}\right) = W_{g_i}\left(\mathbf{u}^{(i)}\right)$; so $\prod_{i \in [k]} B_i\left(v_i, \mathbf{u}^{(i)}\right) = \prod_{i \in \mathsf{supp}(\mathbf{v})} W_{g_i}\left(\mathbf{u}^{(i)}\right)$. From this, we get the required result. □

Suppose in Theorem 28, the g_i's are all equal, i.e. $g_1 = \cdots = g_k = g$. Then $f \circ \mathscr{G} = f \diamond g$ and Theorem 28 provides the Walsh transform of disjoint composition in the case where g is balanced. In this case, the min-entropy is given by the following result.

Theorem 29 *Let k and l be positive integers, f be a k-variable Boolean function, and g be an l-variable balanced Boolean function. For $0 \leq i \leq k$, let $a_i = \max_{\{\mathbf{w}: \mathsf{wt}(\mathbf{w}) = i\}} W_f^2(\mathbf{w})$. Then*

$$H_\infty(f \diamond g) = \min_{i \in \{0, \ldots, k\}, a_i > 0} (-\log(a_i) + i \cdot H_\infty(g)).$$

Proof: Let $n = kl$ and $\mathscr{G}$ be the (n, k) vectorial Boolean function $\mathscr{G}(\mathbf{X}) = (g(\mathbf{X}^{(1)}), \ldots, g(\mathbf{X}^{(k)}))$. Then $f \diamond g = f \circ \mathscr{G}$ and we can apply Theorem 28 to obtain the Walsh transform of $f \diamond g$.

We have from Theorem 28, $W_{f\diamond g}(\mathbf{0}_n) = W_f(\mathbf{0}_k)$, and for $\mathbf{0}_n \neq \mathbf{u} \in \mathbb{F}_2^n$,

$$W_{f\diamond g}(\mathbf{u}) = W_f(\mathbf{w_u}) \prod_{j\in \mathsf{supp}(\mathbf{w_u})} W_g\left(\mathbf{u}^{(j)}\right).$$

From (2.11), to obtain the min-entropy of $f \diamond g$, it is required to obtain $\max_{\mathbf{u}\in\mathbb{F}_2^n}(W_{f\diamond g}(\mathbf{u}))^2$. Let $\boldsymbol{\alpha}_i = \arg\max_{\mathsf{wt}(\mathbf{w})=i} W_f^2(\mathbf{w})$ for $i \in [k]$ and let $\boldsymbol{\beta} = \arg\max_{\mathbf{v}} W_g^2(\mathbf{v})$ (breaking ties arbitrarily in both cases). Note that $a_i = W_f^2(\boldsymbol{\alpha}_i)$ and $H_\infty(g) = -\log W_g^2(\boldsymbol{\beta})$. For $\mathsf{wt}(\mathbf{w_u}) = i$, the maximum value of $\prod_{j\in\mathsf{supp}(\mathbf{w_u})} W_g^2\left(\mathbf{u}^{(j)}\right)$ is $\left(W_g^2(\boldsymbol{\beta})\right)^i$. So $\max_{\mathbf{0}_n\neq\mathbf{u}\in\mathbb{F}_2^n}(W_{f\diamond g}(\mathbf{u}))^2$ is equal to $\max_{i\in[k]} W_f^2(\boldsymbol{\alpha}_i)\left(W_g^2(\boldsymbol{\beta})\right)^i = \max_{i\in[k]} a_i\left(W_g^2(\boldsymbol{\beta})\right)^i$. The result now follows by taking logarithms. □

6.4 Recursive constructions

We wish to obtain a Boolean function f such that $H_\infty(f)/\mathsf{inf}(f)$ is as high as possible. One way to obtain f is to perform an exhaustive search. Since the number of n-variable Boolean functions is 2^{2^n}, it is difficult to carry out the search for $n > 5$. For $n = 5$, we have performed an exhaustive search. This resulted in 3840 5-variable Boolean functions for which the min-entropy/influence ratio is 16/7. All the 3840 functions turned out to be unbalanced. For the purpose of illustration, we provide one of the 3840 functions that were obtained.

Example 1 *Let h be the following 5-variable Boolean function.*

$$h(X_5, X_4, X_3, X_2, X_1) \;=\; X_4X_3 \oplus X_5X_2 \oplus X_5X_4X_1 \oplus X_5X_4X_2 \oplus X_5X_4X_3. \qquad (6.7)$$

For h defined in (6.7)*, $H_\infty(h) = 4$, $\mathsf{inf}(h) = 7/4$ and so $H_\infty(h)/\mathsf{inf}(h) = 16/7$.*

The question now is whether it is possible to obtain a function whose min-entropy/influence ratio is greater than 16/7? In this section, we describe the approaches based on recursive constructions which did not provide such a function. In the next section, we describe a method which yields a function whose min-entropy/influence ratio is greater than that of h.

6.4.1 O'Donnell and Tan's Construction

We first consider the recursive construction of Boolean functions arising from the O'Donnell-Tan construction since this construction proved to be useful for the entropy/influence ratio.

Using Theorem 29, we obtain the following result on the min-entropy of the O'Donnell-Tan recursive construction where the initial function satisfies the condition that there is a vector of weight 1 for which the corresponding Walsh transform value is the maximum.

Theorem 30 *Let g be an l-variable balanced Boolean function for which there is a $\boldsymbol{\beta} \in \mathbb{F}_2^l$ with $\mathsf{wt}(\boldsymbol{\beta}) = 1$ such that $W_g^2(\boldsymbol{\beta}) = \max_{\mathbf{v}} W_g^2(\mathbf{v})$. For $m \geq 0$, let f_m be the Boolean function constructed using (6.3) with $f_0 = g$. Then for $m \geq 0$,*

$$H_\infty(f_m) \;=\; (m+1) \cdot H_\infty(g). \tag{6.8}$$

Consequently,

$$\frac{H_\infty(f_m)}{\mathsf{inf}(f_m)} = \left(\frac{H_\infty(g)}{\mathsf{inf}(g)}\right)\left(\frac{m+1}{\mathsf{inf}(g)^m}\right). \tag{6.9}$$

Proof: Note that $H_\infty(g) = -\log(W_g^2(\boldsymbol{\beta}))$. Further, since g is balanced, using Theorem 28, it follows that f_m is balanced for all $m \geq 1$.

We prove (6.8) by induction on m. For $m = 0$, this follows from the given condition on g. Suppose (6.8) holds for some $m \geq 0$. From Theorem 29 and the fact that f_{m+1} is balanced, $H_\infty(f_{m+1}) = H_\infty(g \diamond f_m) = \min_{i \in [l], a_i > 0}(-\log(a_i) + i \cdot H_\infty(f_m))$, where $a_i = \max_{\mathsf{wt}(\mathbf{w})=i} W_g^2(\mathbf{w})$ for $i = 1, \ldots, l$. For any $i \in [l]$, we have $-\log(a_i) + i \cdot H_\infty(f_m) \geq -\log(W_g^2(\boldsymbol{\beta})) + H_\infty(f_m) = H_\infty(g) + H_\infty(f_m)$ and since $\boldsymbol{\beta}$ has weight 1, equality is attained for $i = 1$. So using the induction hypothesis, $H_\infty(f_{m+1}) = \min_{i \in [l], a_i > 0}(-\log(a_i) + i \cdot H_\infty(f_m)) = H_\infty(g) + H_\infty(f_m) = (m+2)H_\infty(g)$.

The proof of (6.9) follows from (6.8) and Theorem 27. □

To use Theorem 30 as an amplifier of min-entropy/influence ratio it is required to obtain $m \geq 1$ such that $H_\infty(f_m)/\mathsf{inf}(f_m) > H_\infty(g)/\mathsf{inf}(g)$ which holds if and only if $\mathsf{inf}(g) < (m+1)^{1/m}$. For $m = 1$, this condition becomes $\mathsf{inf}(g) < 2$ and for higher values of m, the upper bound on $\mathsf{inf}(g)$ is lower. Comparing (6.4) with (6.9), we see that unlike the case of the entropy/influence ratio, increasing m does not necessarily lead to a higher value of the min-entropy/influence ratio. In particular, the nice asymptotic analyses [130, 86] which has been done for the entropy/influence ratio is not applicable to the min-entropy/influence ratio.

To apply Theorem 30, we need an appropriate initial function g. We performed an exhaustive search over all possible 5-variable Boolean functions which satisfy the conditions of Theorem 30. For $m = 1$, we obtained 384 functions such that taking f_0 to be any of these functions leads to a 25-variable Boolean function f_1 with $H_\infty(f_1)/\mathsf{inf}(f_1) = 512/225 \approx$

2.275556. Let $\mathcal{F}_5$ denote the set of these 384 functions. As an example, we provide one element of $\mathcal{F}_5$.

Example 2 *Let g be the following 5-variable Boolean function.*

$$\begin{aligned} & g(X_5, X_4, X_3, X_2, X_1) \\ &= X_3X_2X_1 \oplus X_4 \oplus X_4X_1 \oplus X_4X_2 \oplus X_4X_2X_1 \oplus X_4X_3X_1 \oplus X_4X_3X_2 \\ &\quad \oplus X_5 \oplus X_5X_1 \oplus X_5X_2X_1 \oplus X_5X_3 \oplus X_5X_3X_1 \oplus X_5X_3X_2 \oplus X_5X_4 \\ &\quad \oplus X_5X_4X_1 \oplus X_5X_4X_2 \oplus X_5X_4X_3. \end{aligned} \tag{6.10}$$

The function g defined in (6.10) *is in $\mathcal{F}_5$. For g, $H_\infty(g) = 4$, $\mathsf{inf}(g) = 15/8$. Taking $f_0 = g$ and $f_1 = f_0 \diamond f_0$, from Theorem 30 we have $H_\infty(f_1)/\mathsf{inf}(f_1) = 32/15 \times 2/(15/8) = 512/225$.*

We note the following points.

1. The 25-variable function f_1 obtained using the above method is not useful. The 5-variable function h given in (6.7) obtained using exhaustive search has a higher value of the min-entropy/influence ratio.

2. In our search over all 5-variable Boolean functions, considering $m > 1$ did not provide a result better than that obtained for $m = 1$.

3. In Theorem 30, the condition $\mathsf{wt}(\boldsymbol{\beta}) = 1$ is required to obtain the expression for f_m given by (6.8). Considering $\mathsf{wt}(\boldsymbol{\beta}) > 1$, on the other hand, does not seem to lead to a higher value of the min-entropy/influence ratio.

6.4.2 A different recursion

Let g be an l-variable Boolean function. We define a sequence $\{g_m\}_{m \geq 0}$ of Boolean functions as follows.

$$\left.\begin{aligned} g_0 &= g, \\ g_m &= g_{m-1} \diamond g_{m-1} \quad \text{if } m \geq 1. \end{aligned}\right\} \tag{6.11}$$

For $m \geq 0$, g_m is a map from $\mathbb{F}_2^{l2^m}$ to $\mathbb{F}_2$. If we start (6.3) and (6.11) with the same initial function g, then we obtain $f_1 = g_1$, but for $m > 1$, the two sequences are different. More

generally, the sequence defined using (6.11) is not a sub-sequence of the sequence defined using (6.3).

Suppose g is a balanced function. Using Theorem 27, it is possible to show that $H(g_m) = H(g)(1+\mathsf{inf}(g_0))(1+\mathsf{inf}(g_1))\ldots(1+\mathsf{inf}(g_{m-1}))$ and $\mathsf{inf}(g_m) = \mathsf{inf}(g)\mathsf{inf}(g_0)\mathsf{inf}(g_1)\ldots\mathsf{inf}(g_{m-1}) = \mathsf{inf}(g)^{2^m}$. From this it is possible to show that $H(g_m)/\mathsf{inf}(g_m) = (H(g)/(\mathsf{inf}(g)-1))\cdot(1-1/\mathsf{inf}(g)^m)$. So as $m\to\infty$, $H(g_m)/\mathsf{inf}(g_m)$ goes to $H(g)/(\mathsf{inf}(g)-1)$ which is the same limit as that obtained from the O'Donnell-Tan recursion. So the recursion given by (6.11) provides a different way of achieving the same limit for the entropy/influence ratio as that obtained using the O'Donnell-Tan recursion.

Suppose g is such that $W_g^2(\mathbf{v})$ is maximum for some $\mathbf{v}$ of weight 1. Let $\{g_m\}_{m\geq 0}$ be the sequence defined in (6.11) with $g_0 = g$. Then in a manner similar to the proof of Theorem 30 it can be shown that $H_\infty(g_m) = 2^m\cdot H_\infty(g)$. So $H_\infty(g_m)/\mathsf{inf}(g_m) = (H(g)/\mathsf{inf}(g))\cdot(2^m/\mathsf{inf}(g)^{2^m-1})$. For $m=1$, this is the same as the O'Donnell-Tan construction and for $m>1$, it does not lead to any improvement over the O'Donnell-Tan construction. So for the min-entropy/influence ratio, the new recursion does not provide anything better than the O'Donnell-tan construction.

6.5 Construction from palindromic functions

An n-variable Boolean function g can be represented by a bit string of length 2^n in the following manner: for $i\in\{0,\ldots,2^n-1\}$, the i-th bit of the string is $g(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is the n-bit binary representation of i. We will denote the bit string representing g also by g. The reverse of the bit string representation of g is g^r, and g^r is given by $g^r(X_n,\ldots,X_1) = g(1\oplus X_n,\ldots,1\oplus X_1)$. The following simple result relates the Walsh transforms of g and g^r.

Proposition 7 *Let g be an n-variable Boolean function and g^r be another n-variable Boolean function defined as $g^r(X_n,\ldots,X_1) = g(1\oplus X_n,\ldots,1\oplus X_1)$. Then for $\boldsymbol{\alpha}\in\mathbb{F}_2^n$, $W_{g^r}(\boldsymbol{\alpha}) = (-1)^{\mathsf{wt}(\boldsymbol{\alpha})}W_g(\boldsymbol{\alpha})$.*

Given an n-variable Boolean function g, we may construct an $(n+1)$-variable Boolean f function in the following manner. Concatenate the bit string representing g and g^r to obtain a bit string of length 2^{n+1}. This string represents the desired $(n+1)$-variable Boolean function f. The bit string representing f is a palindrome and we call f to be a palindromic function. The following construction is a little more general than the method just described.

For $b \in \mathbb{F}_2$, let

$$\begin{aligned} & g_b(X_{n+1}, X_n, \ldots, X_1) \\ & \quad = (1 \oplus X_{n+1}) g(X_n, \ldots, X_1) \oplus X_{n+1}(b + g(1 \oplus X_n, \ldots, 1 \oplus X_1)). \end{aligned} \tag{6.12}$$

If $b = 0$, then f_0 is the concatenation of g and g^r as described above, and if $b = 1$, then f_1 is the concatenation of g and the complement of g^r. The following result shows the relation between the relevant properties of g and g_b.

Proposition 8 *Let g be an n-variable Boolean function and $b \in \mathbb{F}_2$. Let g_b be the $(n+1)$-variable Boolean function constructed from g and b using* (6.12)*. Then the following holds.*

1. *For $\boldsymbol{\beta} \in \mathbb{F}_2^{n+1}$, where $\boldsymbol{\beta} = (a, \boldsymbol{\alpha})$, with $a \in \mathbb{F}_2$ and $\boldsymbol{\alpha} \in \mathbb{F}_2^n$,*

$$W_{g_b}(\boldsymbol{\beta}) \;=\; \left(\frac{(1 + (-1)^{b + \mathsf{wt}(\boldsymbol{\beta})})}{2} \right) W_g(\boldsymbol{\alpha}). \tag{6.13}$$

2. *$H_\infty(g_b) = H_\infty(g)$.*
3. *$\mathsf{inf}(g_b) = \mathsf{inf}(g) + \epsilon_b(g)$, where $\epsilon_b(g) = \displaystyle\sum_{\substack{\boldsymbol{\alpha} \in \mathbb{F}_2^n \\ \mathsf{wt}(\boldsymbol{\alpha}) \not\equiv b \bmod 2}} W_g^2(\boldsymbol{\alpha})$.*

Proof: By definition

$$W_{g_b}(\boldsymbol{\beta}) \;=\; \frac{1}{2^{n+1}} \sum_{\mathbf{x} \in \mathbb{F}_2^{n+1}} (-1)^{g_b(\mathbf{x}) \oplus \langle \boldsymbol{\beta}, \mathbf{x} \rangle}. \tag{6.14}$$

We simplify the exponent in the sum.

$$\begin{aligned} & g_b(x_{n+1}, x_n, \ldots, x_1) \oplus \langle (a, \boldsymbol{\alpha}), (x_{n+1}, x_n, \ldots, x_1) \rangle \\ & \quad = \; (1 \oplus x_{n+1}) g(x_n, \ldots, x_1) \oplus x_{n+1}(b \oplus g(1 \oplus x_n, \ldots, 1 \oplus x_1)) \oplus \langle (a, \boldsymbol{\alpha}), (x_{n+1}, x_n, \ldots, x_1) \rangle \\ & \quad = \; \begin{cases} g(x_n, \ldots, x_1) \oplus \langle \boldsymbol{\alpha}, (x_n, \ldots, x_1) \rangle & \text{if } x_{n+1} = 0, \\ b \oplus g(1 \oplus x_n, \ldots, 1 \oplus x_1) \oplus a \oplus \langle \boldsymbol{\alpha}, (x_n, \ldots, x_1) \rangle & \text{if } x_{n+1} = 1. \end{cases} \end{aligned} \tag{6.15}$$

Writing $\mathbf{x} = (x_{n+1}, \mathbf{y})$, where $x_{n+1} \in \mathbb{F}_2$ and $\mathbf{y} \in \mathbb{F}_2^n$, we simplify (6.14) using (6.15) as follows.

$$W_{g_b}(\boldsymbol{\beta}) \;=\; \frac{1}{2^{n+1}} \left(\sum_{\mathbf{y} \in \mathbb{F}_2^n} (-1)^{g(\mathbf{y}) \oplus \langle \boldsymbol{\alpha}, \mathbf{y} \rangle} + (-1)^{a \oplus b} \sum_{\mathbf{y} \in \mathbb{F}_2^n} (-1)^{g(\mathbf{1}_n \oplus \mathbf{y}) \oplus \langle \boldsymbol{\alpha}, \mathbf{y} \rangle} \right)$$

$$\begin{aligned} &= \frac{1}{2}\left(W_g(\boldsymbol{\alpha}) + (-1)^{a\oplus b} W_{g^r}(\boldsymbol{\alpha})\right) \\ &= \frac{1}{2}\left(W_g(\boldsymbol{\alpha}) + (-1)^{a\oplus b}(-1)^{\mathsf{wt}(\boldsymbol{\alpha})} W_g(\boldsymbol{\alpha})\right) \quad \text{(using Proposition 7)} \\ &= \frac{1}{2}\left(W_g(\boldsymbol{\alpha}) + (-1)^{b}(-1)^{\mathsf{wt}(a,\boldsymbol{\alpha})} W_g(\boldsymbol{\alpha})\right). \end{aligned}$$

This proves the first point. The second point follows directly from the first.

For the third point, we use (2.19) to compute the influence of g_b from its Walsh transform.

$$\begin{aligned} \mathsf{inf}(g_b) &= \sum_{a\in\mathbb{F}_2, \boldsymbol{\alpha}\in\mathbb{F}_2^n} \mathsf{wt}(a,\boldsymbol{\alpha}) W_{g_b}^2(a,\boldsymbol{\alpha}) \\ &= \sum_{a\in\mathbb{F}_2, \boldsymbol{\alpha}\in\mathbb{F}_2^n} \mathsf{wt}(a,\boldsymbol{\alpha}) \left(\frac{(1+(-1)^{b+\mathsf{wt}(a,\boldsymbol{\alpha})})}{2}\right)^2 W_g^2(\boldsymbol{\alpha}) \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \mathsf{wt}(\boldsymbol{\alpha}) \left(\frac{(1+(-1)^{b+\mathsf{wt}(\boldsymbol{\alpha})})}{2}\right)^2 W_g^2(\boldsymbol{\alpha}) \\ &\quad + \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} (1+\mathsf{wt}(\boldsymbol{\alpha})) \left(\frac{(1-(-1)^{b+\mathsf{wt}(\boldsymbol{\alpha})})}{2}\right)^2 W_g^2(\boldsymbol{\alpha}) \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n, \mathsf{wt}(\boldsymbol{\alpha})\equiv b \bmod 2} \mathsf{wt}(\boldsymbol{\alpha}) W_g^2(\boldsymbol{\alpha}) \\ &\quad + \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n, \mathsf{wt}(\boldsymbol{\alpha})\not\equiv b \bmod 2} (1+\mathsf{wt}(\boldsymbol{\alpha})) W_g^2(\boldsymbol{\alpha}) \\ &= \mathsf{inf}(g) + \epsilon_b(g). \end{aligned}$$

□

We note the following two points.

1. The Walsh transform of g_b is banded, i.e. it is zero for all vectors of weights congruent to $1-b$ modulo two.
2. From Parseval's theorem it follows that $0 \leq \epsilon_b(g) \leq 1$.

We recall two well known classes of Boolean function. See [36] for an extensive discussion on the various properties of these classes. Let f be an n-variable Boolean function.

- f is said to be t-resilient, $0 \leq t < n$, if $W_f(\boldsymbol{\alpha}) = 0$ for all $\boldsymbol{\alpha}$ with $\mathsf{wt}(\boldsymbol{\alpha}) \leq t$.
- f is said to be plateaued, if $W_f(\boldsymbol{\alpha})$ takes the values $0, \pm c$, for some c.

From (2.19), it follows that if f is t-resilient, then $\mathsf{inf}(f) \geq t+1$.

Next we present the main result of the chapter.

Theorem 31 *Let g be a balanced n-variable Boolean function, $b \in \mathbb{F}_2$ and g_b be constructed from g and b as in* (6.12)*. Let $G_b = g_b \diamond g$. Then*

$$\frac{H_\infty(G_b)}{\mathsf{inf}(G_b)} = \frac{\min_{i\in\{0,\ldots,k\},a_i>0}(-\log(a_i) + iH_\infty(g))}{\mathsf{inf}(g)(\mathsf{inf}(g) + \epsilon_b(g))}, \tag{6.16}$$

where $a_i = \max_{\{\mathbf{w}:\mathsf{wt}(\mathbf{w})=i\}} W_{g_b}^2(\mathbf{w})$, $i = 0, \ldots, k$.

Further, suppose that there is a $t \geq 0$ such that $t \equiv b \bmod 2$ and g is a plateaued t-resilient function, which is not $(t+1)$-resilient. Then

$$\frac{H_\infty(G_b)}{\mathsf{inf}(G_b)} = \frac{H_\infty(g)}{\mathsf{inf}(g)}\left(\frac{t+3}{\mathsf{inf}(g) + \epsilon_b(g)}\right). \tag{6.17}$$

Proof: Proposition 8 provides the expression for $\mathsf{inf}(g_b)$ and Theorem 27 provides the expression for $\mathsf{inf}(G_b)$. The expression for $H_\infty(G_b)$ is obtained from Theorem 29. This shows (6.16).

Now suppose g is a t-resilient plateaued function such that $t \equiv b \bmod 2$. Since g is plateaued, from (6.13), it follows that g_b is also plateaued and for $a_i > 0$, $-\log(a_i) = H_\infty(g)$. From the conditions g is t-resilient and $t \equiv b \bmod 2$, it follows that g_b is $(t+1)$-resilient. To see this, suppose $\boldsymbol{\beta} \in \mathbb{F}_2^{n+1}$ with $\mathsf{wt}(\boldsymbol{\beta}) \leq t+1$. If $\mathsf{wt}(\boldsymbol{\beta}) = t+1$, then since $t \equiv b \bmod 2$, we have $1 + (-1)^{b+\mathsf{wt}(\boldsymbol{\beta})} = 0$ and so $W_{g_b}(\boldsymbol{\beta}) = 0$; on the other hand, if $\mathsf{wt}(\boldsymbol{\beta}) < t+1$, then writing $\boldsymbol{\beta} = (a, \boldsymbol{\alpha})$ with $a \in \mathbb{F}_2$ and $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, and using the fact that g is t-resilient, it follows that $\mathsf{wt}(\boldsymbol{\alpha}) \leq t$ and so $W_g(\boldsymbol{\alpha}) = 0$ which implies that $W_{g_b}(\boldsymbol{\beta}) = 0$. Further, since g is not $(t+1)$-resilient, it follows that g_b is not $(t+2)$-resilient. Since g_b is $(t+1)$-resilient, but not $(t+2)$-resilient, it follows that the minimum value of i such that $a_i > 0$ is $t+2$. Now using the fact that for $a_i > 0$, $-\log(a_i) = H_\infty(g)$, we have $\min_{i\in\{0,\ldots,k\},a_i>0}(-\log(a_i) + iH_\infty(g)) \geq H_\infty(g) + (t+2)H_\infty(g) = (t+3)H_\infty(g)$. This shows (6.17). □

6.5.1 Construction of a 30-variable Boolean function

By construction, if g is an n-variable Boolean function, then the function G_b in Theorem 31 is an $n(n+1)$-variable Boolean function. To use Theorem 31 as an amplifier of min-entropy/influence ratio, it is required to have $H_\infty(G_b)/\mathsf{inf}(G_b) > H_\infty(g)/\mathsf{inf}(g)$. If g is a

plateaued t-resilient function, then the last condition holds if and only if $t+3 \geq \mathsf{inf}(g)+\epsilon_b(g)$. Note, however, that $\mathsf{inf}(g) \geq t+1$ and so the condition $t+3 \geq \mathsf{inf}(g)+\epsilon_b(g)$ offers only a limited scope for amplification of the min-entropy/influence ratio.

If g is balanced, but not 1-resilient, i.e. $t=0$, then the amplification factor in Theorem 31 is $3/(\mathsf{inf}(g)+\epsilon_b(g))$. We compare this condition with the amplification factor for $m=1$ arising from the O'Donnell-Tan construction. From Theorem 30, the amplification factor in the O'Donnell-Tan construction is $2/\mathsf{inf}(g)$. So if we use the same g in both Theorems 30 and 31, then the amplification provided by Theorem 31 is greater if and only if $\mathsf{inf}(g) > 2\epsilon_b(g)$. The last condition holds for all $g \in \mathcal{F}_5$ (for the definition of $\mathcal{F}_5$ see the discussion before Example 2). So if we take any of the functions in $\mathcal{F}_5$ as the initial function and apply Theorem 31, we will obtain a function whose min-entropy/influence ratio is greater than what can be obtained by starting with the same initial function and using one step of the O'Donnell-Tan construction.

As a concrete example, we consider the 5-variable function g given in Example 2. Using this g and taking $b=0$, from (6.12), we obtain a 6-variable function g_0. The function $G_0 = g_0 \diamond g$ is a 30-variable function. From Theorem 31, we have $H_\infty(G_0)/\mathsf{inf}(G_0) = 128/45 \approx 2.8444$. Starting with any of the 384 functions in $\mathcal{F}_5$ and applying Theorem 31, we obtain a corresponding 30-variable function for which the min-entropy/influence ratio is also 128/45. This gives us a set of 384 30-variable functions each of which has min-entropy/influence ratio to be 128/45. Note that 128/45 is greater than 16/7, which is the maximum min-entropy/influence ratio that is achieved by any 5-variable function (see Example 1 and the discussion preceeding it). Presently, 128/45 is the highest known value of min-entropy/influence ratio that has been achieved. Correspondingly, 128/45 is also the best known lower bound on the universal constant of the min-entropy/influence conjecture.

6.6 Some further search results

A Boolean function f is said to be symmetric if it is invariant under any permutation of its input. The number of n-variable symmetric Boolean functions is 2^{n+1}. O'Donnell et al. [131] established the FEI conjecture for symmetric Boolean functions which also settles the FMEI conjecture for this class of functions. Their proof showed that the entropy/influence ratio of any symmetric Boolean function is at most 12.04. We used exhaustive search to find the actual value of the ratio for symmetric functions on n variables with $n \leq 16$.

For $n \geq 2$, let $A_n(X_1, \ldots, X_n) = X_1 \cdots X_n$ (in terms of Boolean algebra A_n is the AND function). It is easy to show (see [86]) that $H(A_n)/\mathsf{inf}(A_n) < 4$. Our search for $n \leq 16$ showed that if f is an n-variable symmetric Boolean function, then $H(f)/\mathsf{inf}(f) \leq H(A_n)/\mathsf{inf}(A_n)$. This suggests that the ratio 12.04 that was achieved in the proof of [131] is perhaps not the minimum possible value of the entropy/influence ratio for symmetric functions.

A Boolean function f is said to be bent [145] if all the Walsh transform values of f are equal. Such functions can exist only if n is even. If f is bent, then $H(f) = H_\infty(f) = n$. Further, $\mathsf{inf}(f) = n/2$ (see [18]). So for a bent function f, $H(f)/\mathsf{inf}(f) = H_\infty(f)/\mathsf{inf}(f) = 2$. Symmetric functions can be bent and the class of symmetric bent functions have been characterised [152, 113]. Our search showed that if n is even, then for any n-variable symmetric Boolean function f, $H_\infty(f)/\mathsf{inf}(f) \leq 2$ and equality is achieved if and only if f is bent; on the other hand, if n is odd, then for any n-variable symmetric Boolean function f, $H_\infty(f)/\mathsf{inf}(f) < 2$.

Based on our observations, we put forth the following conjecture.

Conjecture 2 *Let f be an n-variable symmetric Boolean function. Then*

1. *$H(f)/\mathsf{inf}(f) \leq H(A_n)/\mathsf{inf}(A_n)$ and equality is achieved if and only if f equals A_n.*
2. *If n is even, then $H_\infty(f)/\mathsf{inf}(f) \leq 2$ and equality is achieved if and only if f is bent; if n is odd, then $H_\infty(f)/\mathsf{inf}(f) < 2$*

A closed form expression for the Walsh transform of symmetric Boolean function in terms of binomial coefficients is known [38]. We could not, however, find a way to use this expression to settle the above conjecture. We also tried to apply the techniques from [131] used for showing that the FEI conjecture holds for symmetric Boolean functions to settle Conjecture 2 but were not successful. The main problem is that the various inequalities used in the proof of [131] do not seem to be sufficiently sharp to establish the bounds stated in the above conjecture. As mentioned above, Conjecture 2 has been verified for $1 \leq n \leq 16$. It is possible to experimentally verify the conjecture for additional values of n, but this is unlikely to provide any insight into how to settle the conjecture.

A Boolean function is said to be rotation symmetric if it is invariant under a cyclic shift of its input. It is not known whether the FEI (or the FMEI) conjecture holds for rotation symmetric Boolean functions. See [162] for the number of rotation symmetric Boolean functions on n variables. We could perform an exhaustive search on rotation symmetric Boolean functions for $n \leq 7$. For $n = 6$ and $n = 7$, the maximum values of $H(f)/\mathsf{inf}(f)$ are

3.739764 and 3.804357 respectively; and the maximum values of $H_\infty(f)/\mathsf{inf}(f)$ are 2.168978 and 2.227449 respectively, where the maximums are over all n-variable rotation symmetric Boolean function. Compared to symmetric Boolean functions, we see that the maximum value of the entropy/influence ratio remains below 4, but the maximum value of the min-entropy/influence ratio is greater than 2. Since we could not run the experiment for higher values of n, we are unable to put forward any conjecture for rotation symmetric Boolean functions.

6.7 Concluding remarks

Our work has opened the interesting topic of obtaining lower bounds on the universal constant of the FMEI conjecture. We have provided one method of constructing Boolean functions which provides the presently best known lower bound. A future challenge is to obtain other construction methods which yield functions with a higher value of the min-entropy/influence ratio. It is also interesting to look for sufficiently sharp techniques to settle Conjecture 2. A final open problem resulting from our work is to settle the FEI conjecture for rotation symmetric Boolean functions.

Chapter 7

"Majority is Least stable" conjecture

In this chapter, we present an example which proves that the "majority is least stable" conjecture holds true for $n = 1$ and 3, while being false for all odd $n \geq 5$. The statement of the conjecture was taken verbatim from the book [128] (Page 133), where it is stated for Boolean functions $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$. To ensure consistency, we have decided to maintain the same definition of a Boolean function in this chapter. So, a Boolean function in this chapter will look like $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$. Additionally, in order to facilitate understanding, we provide definitions of the Fourier transform, ρ-noisy distribution, noise stability, and influence over the domain $\{-1,1\}^n$, which are equivalent to the definitions provided in Chapter 2 for the domain $\mathbb{F}_2^n$. To maintain clarity and avoid confusion, we use the notations for each of these concepts in the same way as we did in Chapter 2, even though their representations differ.

7.1 Introduction

We start by defining the *linear threshold function* and majority function for the input values in the $\{-1,1\}^n$.

A Boolean function $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ is said to be a *linear threshold function* if there are real constants $w_0, w_1, \ldots, w_n$ such that for any $\mathbf{x} = (x_1, \ldots, x_n) \in \{-1,1\}^n$, $\mathfrak{f}(\mathbf{x}) = \mathsf{sgn}(w_0 + w_1x_1 + \cdots + w_nx_n)$, where $\mathsf{sign}(z) = 1$ if $z \geq 0$, and -1 if $z < 0$.

For odd n, the majority function $\mathsf{Maj}_n : \{-1,1\}^n \to \{-1,1\}$ is the following.

$$\mathsf{Maj}_n(x_1, \ldots, x_n) = \mathsf{sign}(x_1 + x_2 + \ldots + x_n).$$

For $\mathbf{x} \in \{-1,1\}^n$ and $\rho \in [0,1]$, the distribution $N_\rho(\mathbf{x})$ over $\{-1,1\}^n$ is defined in the following manner (see Page 53 of [127]): $\mathbf{y} = (y_1, \ldots, y_n) \sim N_\rho(\mathbf{x})$ if for $i = 1, \ldots, n$,

$$y_i = \begin{cases} x_i, & \text{with probability } \rho \\ \pm 1, & \text{with probability } (1-\rho)/2 \text{ each.} \end{cases}$$

The noise stability of a function $\mathfrak{f} : \{-1,1\}^n \to \mathbb{R}$, denoted by $\mathsf{Stab}_\rho(\mathfrak{f})$, is defined as follows.

$$\mathsf{Stab}_\rho(\mathfrak{f}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim\{-1,1\}^n,\, \mathbf{y}\sim N_\rho(\mathbf{x})} [\mathfrak{f}(\mathbf{x})\mathfrak{f}(\mathbf{y})]\,. \tag{7.1}$$

Benjamini, Kalai and Schramm in 1999 (see [13, 61]) put forward the following conjecture.

Conjecture 3 (*"Majority is Least Stable"*) : *Let n be odd and $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ be a* linear threshold function. *Then for all $\rho \in [0,1]$, $\mathsf{Stab}_\rho(\mathfrak{f}) \geq \mathsf{Stab}_\rho(\mathsf{Maj}_n)$.*

Counterexamples to this conjecture have been found for $n = 5$ by Noam Berger and Vishesh Jain (see [139, 127, 89]), and for every odd n by Daniel Kane and Steven Heilman, as well as independently by Sivakanth Gopi (private communication). These counterexamples, however, have never been made public. To the best of our knowledge, the only public counterexample to the conjecture was published by Jain [89] for the case of $n = 5$.

In this chapter, we show that Conjecture 3 is true for $n = 1$ and 3 and false for odd $n \geq 5$. To show that the conjecture is false for odd $n \geq 5$, we define a sequence of Boolean functions $\mathfrak{g}_n$ and show that $\mathsf{Stab}_\rho(\mathfrak{g}_n) < \mathsf{Stab}_\rho(\mathsf{Maj}_n)$. To show that the conjecture is true for $n = 3$, we employed a search over all *monotone* 3-variable Boolean functions $\mathfrak{f}$ and obtained the expressions for $\mathsf{Stab}_\rho(\mathfrak{f})$. It turns out that each of these expressions is greater than or equal to $\mathsf{Stab}_\rho(\mathsf{Maj}_n)$ for all $\rho \in [0,1]$.

7.2 Preliminaries

For $S \subseteq [n]$, the characters of $\{-1,1\}^n$, $\mathfrak{C}_S : \{-1,1\}^n \to \{-1,1\}$ are defined as: $\mathfrak{C}_S(x_1,\ldots,x_n) = \prod_{i\in S} x_i$. Just as in the case of Theorem 1, it is also possible to show that $\{\mathfrak{C}_S\}_{S\subseteq[n]}$ is a collection of orthonormal characters.

Then the Fourier transform of $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$, which is a map $\widehat{\mathfrak{f}} : 2^{[n]} \to [-1,1]$ can be defined as follows. For $S \subseteq [n]$,

$$\widehat{\mathfrak{f}}(S) \;=\; \frac{1}{2^n} \sum_{\mathbf{x}\in\{-1,1\}^n} \mathfrak{f}(\mathbf{x})\mathfrak{C}_S(\mathbf{x}). \tag{7.2}$$

For $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ and $k \in [n]$, let $W^{(k)}[\mathfrak{f}] = \sum_{S\subseteq[n],|S|=k} \widehat{\mathfrak{f}}^2(S)$ and $W^{\leq k}[\mathfrak{f}] =$

$\sum_{i=0}^{k} W^{(i)}[\mathfrak{f}]$. We say that $\mathfrak{f}$ is balanced if $\#\{\mathbf{x} : \mathfrak{f}(\mathbf{x}) = 1\} = \#\{\mathbf{x} : \mathfrak{f}(\mathbf{x}) = -1\}$. It follows that $\mathfrak{f}$ is balanced if and only if $\widehat{\mathfrak{f}}(\emptyset) = 0$.

The Fourier expression of $\mathsf{Stab}_\rho(\mathfrak{f})$ is the following (see Page 56 of [127]). Note that this expression is essentially equivalent to the one presented in Lemma 8.

$$\mathsf{Stab}_\rho(\mathfrak{f}) = \sum_{k=0}^{n} \rho^k \cdot W^{(k)}[\mathfrak{f}]. \tag{7.3}$$

As we have previously observed (see 2.17), the influence of a variable i on a Boolean function is determined by calculating the probability that flipping the value of variable i will result in a change in the function's value. Suppose, for any $\mathbf{x} \in \{-1,1\}^n$, $\mathbf{x}^{\oplus i}$ denotes the vector $(x_1, \ldots, x_{i-1}, -x_i, x_{i+1}, \ldots, x_n)$. Then the notion of influence of a variable $i \in [n]$ over a Boolean function $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$, $\mathsf{inf}_{\mathfrak{f}}(i)$ will be defined as follows (see Page 46 of [127]).

$$\mathsf{inf}_{\mathfrak{f}}(i) = \Pr_{\mathbf{x} \in \{-1,1\}^n}[\mathfrak{f}(\mathbf{x}) \neq \mathfrak{f}(\mathbf{x}^{\oplus \mathbf{i}})].$$

It is easy to see that Maj_n is balanced and so $W^{(0)}[\mathsf{Maj}_n] = 0$. It is known that (see Page 62 of [127])

$$W^{(1)}[\mathsf{Maj}_n] = \left[\frac{\binom{n-1}{\frac{n-1}{2}}}{2^{n-1}}\right]^2 \cdot n. \tag{7.4}$$

It was observed in [89] that if $\mathfrak{f}$ is a balanced *linear threshold function*, then showing $W^{(1)}[\mathfrak{f}] < W^{(1)}[\mathsf{Maj}_n]$ would disprove Conjecture 3. For the sake of completeness, we state a more general form of this observation as a lemma and provide a proof.

Lemma 12 *Let n be odd and $\mathfrak{f} : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function such that $W^{(0)}[\mathfrak{f}] = 0$ and $W^{(1)}[\mathfrak{f}] < W^{(1)}[\mathsf{Maj}_n]$. Then there exists a $\delta > 0$ such that $\mathsf{Stab}_\rho(\mathfrak{f}) < \mathsf{Stab}_\rho(\mathsf{Maj}_n)$ for all $0 < \rho < \delta$. Consequently, the function $\mathfrak{f}$ is a counter-example to Conjecture 3.*

Proof: For $k \geq 0$, let $a_k = W^{(k)}[\mathfrak{f}] - W^{(k)}[\mathsf{Maj}_n]$. Since by assumption, $W^{(0)}[\mathfrak{f}] = 0$, $W^{(1)}[\mathfrak{f}] < W^{(1)}[\mathsf{Maj}_n]$, and noting that Maj_n is balanced, it follows that $a_0 = 0$ and $-1 \leq a_1 < 0$. On the other hand, for $k \geq 2$, we have $-1 \leq a_k < 1$.

Now, $\mathsf{Stab}_\rho(\mathfrak{f}) - \mathsf{Stab}_\rho(\mathsf{Maj}_n) = \sum_{k=1}^{n} \rho^k \cdot a_k$. Therefore, $\mathsf{Stab}_\rho(\mathfrak{f}) - \mathsf{Stab}_\rho(\mathsf{Maj}_n) < 0$ if and only if $\rho(a_2 + \rho a_3 + \ldots + \rho^{n-2} a_n) < -a_1$. Since $a_k < 1$ for $k = 2, \ldots, n$, it follows

that $\rho(a_2 + \rho a_3 + \ldots + \rho^{n-2}a_n)$ is upper bounded by $\rho(1 + \rho + \ldots + \rho^{n-2})$ whose limiting value is 0 as $\rho \to 0$. Therefore, there must exist some $\delta > 0$ such that for all $0 < \rho < \delta$, $\rho(a_2 + \rho a_3 + \ldots + \rho^{n-2}a_n) < -a_1$. Consequently, $\mathsf{Stab}_\rho(\mathfrak{f}) < \mathsf{Stab}_\rho(\mathsf{Maj}_n)$ for all $0 < \rho < \delta$. $\square$

A function $\mathfrak{f}(X_1, \ldots, X_n)$ is *monotone* increasing in $\mathbf{X}_1$, if and only if $\mathfrak{f}(-1, \mathbf{Y}) \leq \mathfrak{f}(1, \mathbf{Y})$, for all possible values of $\mathbf{Y} = (X_2, \ldots, X_n)$ taken from $\{-1, 1\}^{n-1}$. Likewise, it is *monotone* decreasing in X_1 if and only if $\mathfrak{f}(1, \mathbf{Y}) \leq \mathfrak{f}(-1, \mathbf{Y})$. $\mathfrak{f}$ is said to be *unate* or locally *monotone* if it is *monotone* increasing or decreasing in each variable.

From [73] (see Lemma 2.2 and the comment following it), it follows that if $\mathfrak{f}$ is a unate function, then for all $i \in [n]$, $\mathsf{inf}_\mathfrak{f}(i) = |\widehat{\mathfrak{f}}(\{i\})|$. Since a *linear threshold function* is unate, we have the following result which has been used in the proof of Theorem 4.1 of [73].

Theorem 32 (Gotsman and Linial [73]) *If $\mathfrak{f} : \{-1, 1\}^n \to \{-1, 1\}$ is a* linear threshold function*, then $\sum_{i=1}^{n} \left(\mathsf{inf}_\mathfrak{f}(i)\right)^2 = W^{(1)}[\mathfrak{f}]$.*

7.3 Settling Conjecture 3

We state and prove some results from which the main theorem follows.

Lemma 13 *Let $n \geq 1$, w_0 be an integer and w_1 and w_2 be non-zero integers. Let T be a subset of $[n]$ of cardinality $t \leq n/2$. Consider the following* linear threshold function*: $\mathfrak{f}(x_1, \ldots, x_n) = \mathsf{sign}\left(w_0 + w_1 \cdot \sum_{u \in T} x_u + w_2 \cdot \sum_{v \in \overline{T}} x_v\right)$. Then*

$$W^{(0)}[\mathfrak{f}] = \frac{1}{2^{2n}} \cdot \left[\sum_{(i,j) \in \mathcal{S}_0} \binom{t}{i}\binom{n-t}{j}\right]^2, \tag{7.5}$$

$$W^{(1)}[\mathfrak{f}] = \frac{t}{2^{2n-2}} \cdot \left[\sum_{(i,j) \in \mathcal{S}_1} \binom{t-1}{i}\binom{n-t}{j}\right]^2 + \frac{n-t}{2^{2n-2}} \cdot \left[\sum_{(i,j) \in \mathcal{S}_2} \binom{t}{i}\binom{n-t-1}{j}\right]^2 \tag{7.6}$$

where $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are defined as follows.

$$\mathcal{S}_0 = \begin{cases} \{(i,j) : 0 \le i \le t,\ 0 \le j \le n-t \\ \qquad \textit{and } -w_0 \le w_1(2i-t) + w_2(2j-(n-t)) \le w_0\} & \textit{if } w_0 \ge 0, \\ \{(i,j) : 0 \le i \le t,\ 0 \le j \le n-t \\ \qquad \textit{and } w_0 < w_1(2i-t) + w_2(2j-(n-t)) < -w_0\} & \textit{if } w_0 < 0; \end{cases}$$

$$\mathcal{S}_1 = \{(i,j) : 0 \le i \le t-1,\ 0 \le j \le n-t \quad \textit{and } -|w_1| \le w_0 + w_1(2i-(t-1)) + w_2(2j-(n-t)) < |w_1|\};$$

$$\mathcal{S}_2 = \{(i,j) : 0 \le i \le t,\ 0 \le j \le n-t-1 \quad \textit{and } -|w_2| \le w_0 + w_1(2i-t) + w_2(2j-(n-t-1)) < |w_2|\}.$$

Proof: We start with the proof of (7.5). For $\mathbf{x} \in \{-1,1\}^n$, let $A(\mathbf{x}) = w_1 \cdot \sum_{u \in T} x_u + w_2 \cdot \sum_{v \in \overline{T}} x_v$ so that $\mathfrak{f}(\mathbf{x}) = \mathsf{sgn}(w_0 + A(\mathbf{x}))$. Let N (resp. M) be the number of $\mathbf{x}$'s such that $w_0 + A(\mathbf{x}) \ge 0$ (resp. $w_0 + A(\mathbf{x}) < 0$). Then $\widehat{\mathfrak{f}}(\emptyset) = (N-M)/2^n$. There are two cases to consider.

First consider the case $w_0 \ge 0$. Let N_1 (resp. N_2) be the number of $\mathbf{x}$'s such that $A(\mathbf{x}) > w_0$ (resp. $-w_0 \le A(\mathbf{x}) \le w_0$). So, $N = N_1 + N_2$. Since $A(-\mathbf{x}) = -A(\mathbf{x})$, it follows that $N_1 = M$ and so $\widehat{\mathfrak{f}}(\emptyset) = N_2/2^n$. Therefore to obtain $W^{(0)}[\mathfrak{f}] = \widehat{\mathfrak{f}}^2(\emptyset)$ it is sufficient to obtain N_2. For $\mathbf{x} \in \{-1,1\}^n$, let $i = \#\{u \in T : x_u = 1\}$ and $j = \#\{v \in \overline{T} : x_v = 1\}$. Then $A(\mathbf{x}) = w_1(2i-t) + w_2(2j-(n-t))$. For $0 \le i \le t$ and $0 \le j \le n-t$, the pair (i,j) is in $\mathcal{S}_0$ if and only if $-w_0 \le A(\mathbf{x}) \le w_0$. So, the number of $\mathbf{x}$'s for which $-w_0 \le A(\mathbf{x}) \le w_0$ holds is $\sum_{(i,j) \in \mathcal{S}_0} \binom{t}{i}\binom{n-t}{j}$ which is the value of N_2.

Next consider the case $w_0 < 0$. Let M_1 (resp. M_2) be the number of $\mathbf{x}$'s such that $A(\mathbf{x}) \le w_0$ (resp. $w_0 < A(\mathbf{x}) < -w_0$). So, $M = M_1 + M_2$. Again since $A(-\mathbf{x}) = -A(\mathbf{x})$, it follows that $M_1 = N$ and so $\widehat{\mathfrak{f}}(\emptyset) = -M_2/2^n$. Therefore to obtain $W^{(0)}[\mathfrak{f}] = \widehat{\mathfrak{f}}^2(\emptyset)$ it is sufficient to obtain M_2. A similar argument as above shows that M_2 is equal to $\sum_{(i,j) \in \mathcal{S}_0} \binom{t}{i}\binom{n-t}{j}$.

Now we turn to the proof of (7.6). Fix some $s \in T$ and some $r \in \overline{T}$. Due to symmetry, for any $i \in T$, we have $\mathsf{inf}_{\mathfrak{f}}(i) = \mathsf{inf}_{\mathfrak{f}}(s)$ and for any $j \in \overline{T}$, we have $\mathsf{inf}_{\mathfrak{f}}(j) = \mathsf{inf}_{\mathfrak{f}}(r)$ and so from Theorem 32,

$$W^{(1)}[\mathfrak{f}] = t \cdot (\mathsf{inf}_{\mathfrak{f}}(s))^2 + (n-t) \cdot (\mathsf{inf}_{\mathfrak{f}}(r))^2 . \tag{7.7}$$

Let N_s (resp. N_r) be the number of $\mathbf{x} \in \{-1,1\}$ such that $\mathfrak{f}(\mathbf{x}) \neq \mathfrak{f}(\mathbf{x}^{\oplus s})$ (resp. $\mathfrak{f}(\mathbf{x}) \neq \mathfrak{f}(\mathbf{x}^{\oplus r})$). Then $\mathsf{inf}_{\mathfrak{f}}(s) = N_s/2^{n-1}$ and $\mathsf{inf}_{\mathfrak{f}}(r) = N_r/2^{n-1}$.

For $\mathbf{x} \in \{-1,1\}^n$, let $B(\mathbf{x}) = w_0 + w_1 \sum_{u \in T \setminus \{s\}} x_u + w_2 \sum_{v \in \overline{T}} x_v$. From the definition of $\mathfrak{f}$, N_s is the number of $\mathbf{x}$'s such that either $(w_1 x_s + B(\mathbf{x}) \ge 0$ and $-w_1 x_s + B(\mathbf{x}) < 0)$ or $(w_1 x_s +$

$B(\mathbf{x}) < 0$ and $-w_1x_s + B(\mathbf{x}) \geq 0$) holds. The two conditions are equivalent to $-w_1x_s \leq B(\mathbf{x}) < w_1x_s$ and $w_1x_s \leq B(\mathbf{x}) < -w_1x_s$ respectively. For the first condition, we must have $w_1x_s > 0$ as otherwise we obtain $|w_1x_s| \leq B(\mathbf{x}) < -|w_1x_s|$ which is a contradiction; similarly, for the second condition, we must have $w_1x_s < 0$. Noting that $x_s \in \{-1, 1\}$, both conditions boil down to $-|w_1| \leq B(\mathbf{x}) < |w_1|$, and consequently, N_s is the number of $\mathbf{x}$'s such that $-|w_1| \leq B(\mathbf{x}) < |w_1|$ holds.

For $\mathbf{x} \in \{-1, 1\}^n$, let $i = \#\{u \in T \setminus \{s\} : x_u = 1\}$ and $j = \#\{v \in \overline{T} : x_v = 1\}$. Then $B(\mathbf{x}) = w_0 + w_1(2i - (t-1)) + w_2(2j - (n-t))$. For $0 \leq i \leq t-1$ and $0 \leq j \leq n-t$, the pair (i, j) is in $\mathcal{S}_1$ if and only if $-|w_1| \leq B(\mathbf{x}) < |w_1|$ holds. So, the number of $\mathbf{x}$'s for which $-|w_1| \leq B(\mathbf{x}) < |w_1|$ holds is $\sum_{(i,j)\in\mathcal{S}_1} \binom{t-1}{i}\binom{n-t}{j}$ which is the value of N_s.

A similar argument shows that N_r is equal to $\sum_{(i,j)\in\mathcal{S}_2} \binom{t}{i}\binom{n-t-1}{j}$. Using the values of N_s and N_r to obtain $\mathsf{inf}_{\mathfrak{f}}(s)$ and $\mathsf{inf}_{\mathfrak{f}}(r)$ respectively and substituting these in (7.7) gives the expression for $W^{(1)}[\mathfrak{f}]$ stated in (7.6). □

For odd $n \geq 3$, we define a sequence of functions $\mathfrak{g}_n : \{-1, 1\}^n \to \{-1, 1\}$ where

$$\mathfrak{g}_n(x_1, \ldots, x_n) \;=\; \mathsf{sign}(2 \cdot (x_1 + \ldots + x_{n-3}) + x_{n-2} + x_{n-1} + x_n). \tag{7.8}$$

In [89], the function $\mathfrak{g}_5$ has been shown to be a counter-example to Conjecture 3.

Lemma 14 *For $\mathfrak{g}_n$ defined in* (7.8)*, we have*

$$\begin{aligned} W^{(0)}[\mathfrak{g}_n] &= 0, \\ W^{(1)}[\mathfrak{g}_n] &= \begin{cases} 3 \cdot \left[\frac{2}{2^{n-1}}\right]^2, & \text{if } n = 3 \\ (n-3) \cdot \left[\frac{\binom{n-4}{\frac{n-5}{2}} \cdot 8}{2^{n-1}}\right]^2 + 3 \cdot \left[\frac{\binom{n-3}{\frac{n-3}{2}} \cdot 2}{2^{n-1}}\right]^2, & \text{for odd } n \geq 5. \end{cases} \end{aligned} \tag{7.9}$$

Proof: For $n = 3$, it is clear that $\mathfrak{g}_3 = \mathsf{Maj}_3$. Therefore, from the fact that the Maj_3 is balanced and (7.4) we obtain the desired result. Now let us assume that $n > 3$. We use Lemma 13. For $\mathfrak{g}_n$, we have $w_0 = 0$, $w_1 = 1$ and $w_2 = 2$. Also, $T = \{n-2, n-1, n\}$ so that $t = 3$. With these values, the sets $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ defined in Lemma 13 are the following.

$$\begin{aligned} \mathcal{S}_0 &= \{(i,j) : 0 \leq i \leq 3,\, 0 \leq j \leq n-3 \text{ and } (2i-3) + 2(2j - (n-3)) = 0\}, \\ \mathcal{S}_1 &= \{(i,j) : 0 \leq i \leq 2,\, 0 \leq j \leq n-3 \text{ and } -1 \leq (2i-2) + 2(2j - (n-3)) < 1\}, \\ \mathcal{S}_2 &= \{(i,j) : 0 \leq i \leq 3,\, 0 \leq j \leq n-4 \text{ and } -2 \leq (2i-3) + 2(2j - (n-4)) < 2\}. \end{aligned}$$

Since $(2i-3) + 2(2j - (n-3))$ is odd, it cannot be zero and so $\mathcal{S}_0$ is empty showing that

$W^{(0)}[\mathfrak{g}_n] = 0$.

Since $(2i-2)+2(2j-(n-3))$ is even it cannot be equal to -1 and so the only possible value it can take is 0. From this, we obtain $\mathcal{S}_1$ to be $\{(1,(n-3)/2)\}$.

Similarly, since $(2i-3)+2(2j-(n-4))$ is odd, the only possible values in the set $\{-2,-1,0,1\}$ that it can take are -1 and 1. Corresponding to these two values, we obtain $j=(n-3-i)/2$ and $j=(n-2-i)/2$ respectively. Since n is odd, in the first case i must be even, while in the second case i must be odd. So, $\mathcal{S}_2 = \{(0,(n-3)/2),(2,(n-5)/2),(1,(n-3)/2),(3,(n-5)/2)\}$.

Substituting the values of w_0, w_1, w_2, t as well as $\mathcal{S}_1$ and $\mathcal{S}_2$ in (7.6), we obtain

$$W^{(1)}[\mathfrak{g}_n] = \frac{3}{2^{2n-2}} \cdot \left[\binom{2}{1} \binom{n-3}{\frac{n-3}{2}} \right]^2 + \frac{n-3}{2^{2n-2}} \cdot \left[\binom{n-4}{\frac{n-3}{2}} + 3\binom{n-4}{\frac{n-5}{2}} + 3\binom{n-4}{\frac{n-3}{2}} + \binom{n-4}{\frac{n-5}{2}} \right]^2$$

Noting that $(n-3)/2+(n-5)/2 = n-4$ leads to the expression for $W^{(1)}[\mathfrak{g}_n]$ given in (7.9). $\square$

Lemma 15 *Let $\mathfrak{g}_n$ be defined as in* (7.8)*. For odd $n \geq 5$, $W^{(1)}[\mathfrak{g}_n] < W^{(1)}[\mathsf{Maj}_n]$.*

Proof: The expression for $W^{(1)}[\mathsf{Maj}_n]$ is given by (7.4) and the expression for $W^{(1)}[\mathfrak{g}_n]$ is given by (7.9). Therefore

$$\frac{W^{(1)}[\mathfrak{g}_n]}{W^{(1)}[\mathsf{Maj}_n]} = \left[\frac{\binom{n-4}{\frac{n-5}{2}} \cdot 8}{\binom{n-1}{\frac{n-1}{2}}} \right]^2 \left(\frac{n-3}{n} \right) + \left[\frac{\binom{n-3}{\frac{n-3}{2}} \cdot 2}{\binom{n-1}{\frac{n-1}{2}}} \right]^2 \left(\frac{3}{n} \right) = \left[\frac{n-1}{n-2} \right]^2 \left(\frac{4n-9}{4n} \right). \tag{7.10}$$

From (7.10), it follows that $W^{(1)}[\mathfrak{g}_n] < W^{(1)}[\mathsf{Maj}_n]$ if and only if $(n-3)^2 > 0$ i.e. $n \geq 5$. $\square$

Lemma 16 *Conjecture 3 is true for $n = 1$ and $n = 3$.*

Proof: For $n = 1$, the only *linear threshold function* (ltf) is the majority function and so Conjecture 3 is trivially true.

Using (7.3), for $n = 3$, it is easy to check that $\mathsf{Stab}_\rho(\mathsf{Maj}_3) = 0.75\rho + 0.25\rho^3$. We need to compare this expression with $\mathsf{Stab}_\rho(\mathsf{f})$ where f is an ltf. We used an exhaustive search. There is no easy way to determine whether a given function is an ltf. Since an ltf is unate, if we

search over all 3-variable unate functions the search will cover all ltfs. From the definition of noise stability given in (7.1), it follows that negating some of the inputs of an ltf does not affect the noise stability. Since by negating inputs an ltf can be converted to a monotone function, it suffices to search over all 3-variable monotone functions. Let $\mathfrak{f}$ be a 3-variable monotone function. We obtained the values of $W^{(k)}[\mathfrak{f}]$, for $k = 0, 1, 2, 3$ and using (7.3) obtained the expression for $\mathsf{Stab}_\rho(\mathfrak{f})$. From the search, the possible expressions for $\mathsf{Stab}_\rho(\mathfrak{f})$ were obtained to be the following: 1, ρ, $0.75\rho+0.25\rho^3$, $0.0625+0.6875\rho+0.1875\rho^2+0.0625\rho^3$, $0.25 + 0.5\rho + 0.25\rho^2$, $0.5625 + 0.1875\rho + 0.1875\rho^2 + 0.0625\rho^3$. For each of these expressions, it is easy to verify that $\mathsf{Stab}_\rho(\mathfrak{f}) \geq \mathsf{Stab}_\rho(\mathsf{Maj}_3)$ for all $\rho \in [0, 1]$. □

Based on Lemmas 12, 15 and 16, we obtain the main result of the paper, of which the case $n = 5$ was reported in [89].

Theorem 33 *Conjecture 3 is true for $n = 1$ and $n = 3$. For odd $n \geq 5$, Conjecture 3 is false.*

For $n \geq 5$, there are other functions which provide counterexamples to Conjecture 3. For odd n, suppose $\mathfrak{h}_n$ is obtained from $\mathfrak{g}_n$ by negating some of the input variables. This does not affect the noise stability, i.e. $\mathsf{Stab}_\rho(\mathfrak{h}_n) = \mathsf{Stab}_\rho(\mathfrak{g}_n)$. So for odd $n \geq 5$, the function $\mathfrak{h}_n$ is also a counterexample to Conjecture 3.

7.4 Limiting Value of $W^{\leq 1}[g_n]$

It is known (see Page 62 of [127]) that $W^{\leq 1}[\mathsf{Maj}_n]$ is a decreasing sequence which is lower bounded by $2/\pi$. In the context being discussed, there exists another reasonable conjecture that is referenced in both [13] and on Page 115 of [127], which is as follows.

Conjecture 4 *If $\mathfrak{f} : \{-1, 1\}^n \to \{-1, 1\}$ is a* linear threshold function*, then $W^{\leq 1}[\mathfrak{f}] \geq \frac{2}{\pi}$.*

We have shown that for odd $n \geq 5$, the function $\mathfrak{g}_n$ defined by (7.8) satisfies $W^{(1)}[\mathfrak{g}_n] < W^{(1)}[\mathsf{Maj}_n]$. This brings up the question of whether the sequence $\mathfrak{g}_n$ also provides a counterexample to the Conjecture 4. In this section, we show that this is not the case.

From Lemma 14, $W^{(0)}[\mathfrak{g}_n] = 0$ and so, $W^{\leq 1}[\mathfrak{g}_n] = W^{(1)}[\mathfrak{g}_n]$. This shows that it is sufficient to consider $W^{(1)}[\mathfrak{g}_n]$. We show that $W^{(1)}[\mathfrak{g}_n]$ is a decreasing sequence which is lower bounded by $2/\pi$. The expression for $W^{(1)}[\mathfrak{g}_n]$ given by (7.9) involves binomial coefficients. We use

the following bounds on factorial function (see Page 54 of [59]).

$$\sqrt{2\pi m}\cdot\frac{m^m}{e^m}\exp\left(\frac{1}{12m+1}\right)\leq m!\leq\sqrt{2\pi m}\cdot\frac{m^m}{e^m}\exp\left(\frac{1}{12m}\right). \tag{7.11}$$

Let $p=\frac{k}{m}$ and $q=1-p$. Using (7.11), the following bounds on $\binom{m}{k}$ can be obtained.

$$\left.\begin{array}{rcl}\binom{m}{k} & \geq & \frac{1}{\sqrt{2\pi mpq}}(p^pq^q)^{-m}\exp\left(\frac{1}{12m+1}-\frac{1}{12k}-\frac{1}{12(m-k)}\right), \\ \binom{m}{k} & \leq & \frac{1}{\sqrt{2\pi mpq}}(p^pq^q)^{-m}\exp\left(\frac{1}{12m}-\frac{1}{12k+1}-\frac{1}{12(m-k)+1}\right).\end{array}\right\} \tag{7.12}$$

Lemma 17 *For $\mathfrak{g}_n$ defined in (7.8), $W^{(1)}[\mathfrak{g}_n]$ is a decreasing sequence and $\lim_{n\to\infty}W^{(1)}[\mathfrak{g}_n]=\frac{2}{\pi}$. Consequently, for all odd n, $W^{(1)}[\mathfrak{g}_n]\geq 2/\pi$.*

Proof: Let $a_n=W^{(1)}[\mathfrak{g}_n]$ and $b_n=W^{(1)}[\mathsf{Maj}_n]$. We wish to show that a_n is a decreasing sequence. To do this, we compare a_{n+2}/b_n to a_n/b_n. The expression for a_n/b_n is given by (7.10). Using (7.4) and (7.9), we obtain $a_{n+2}/b_n=(4n-1)/(4n)$. We have $a_n>a_{n+2}$ if and only if $a_n/b_n>a_{n+2}/b_n$. Using the expressions for a_n/b_n and a_{n+2}/b_n, the last condition is equivalent to

$$\left[\frac{n-1}{n-2}\right]^2\left(\frac{4n-9}{4n}\right)>\frac{4n-1}{4n}$$

which holds if and only if $n\geq 3$. So a_n is a decreasing sequence for all odd $n\geq 3$.

Let $A_n=(n-3)\cdot\left[\frac{\binom{n-4}{\frac{n-5}{2}}\cdot 8}{2^{n-1}}\right]^2$, $B_n=3\cdot\left[\frac{\binom{n-3}{\frac{n-3}{2}}\cdot 2}{2^{n-1}}\right]^2$ and so $a_n=A_n+B_n$. We show that A_n tends to $2/\pi$ and B_n tends to 0 and so a_n tends to $2/\pi$ as n goes to infinity.

First consider A_n. Letting $m=n-4$, $k=\frac{n-5}{2}$, $p=k/m$ and $q=1-p$, from (7.12) and using some routine simplifications we obtain the following bounds on A_n.

$$\begin{array}{rcl}A_n & \geq & \frac{2}{\pi}\left[\frac{n-4}{n-3}\right]^{(n-3)}\left[\frac{n-5}{n-4}\right]^{-(n-4)}\exp\left(\frac{2}{12n-47}-\frac{2}{6n-30}-\frac{2}{6n-18}\right), \\ A_n & \leq & \frac{2}{\pi}\left[\frac{n-4}{n-3}\right]^{(n-3)}\left[\frac{n-5}{n-4}\right]^{-(n-4)}\exp\left(\frac{2}{12n-48}-\frac{2}{6n-29}-\frac{2}{6n-17}\right).\end{array}$$

Since $\lim_{x\to\infty}(1-\frac{1}{x})^x=\frac{1}{e}$ and $\lim_{x\to\infty}(1-\frac{1}{x})^{-x}=e$, it follows that $\lim_{n\to\infty}A_n=2/\pi$.

Now, consider B_n. Letting $m=n-3$, $k=\frac{n-3}{2}$ and $p=q=\frac{1}{2}$, from (7.12) and using

some routine simplifications we obtain the following bounds on B_n.

$$\begin{aligned} B_n &\geq \frac{3}{2\pi}\left(\frac{1}{n-3}\right)\exp\left(\frac{2}{12n-35}-\frac{2}{6n-18}-\frac{2}{6n-18}\right), \\ B_n &\leq \frac{3}{2\pi}\left(\frac{1}{n-3}\right)\exp\left(\frac{2}{12n-36}-\frac{2}{6n-17}-\frac{2}{6n-17}\right). \end{aligned}$$

It follows that $\lim_{n\to\infty} B_n = 0$. □

Chapter 8

Counting unate and balanced monotone Boolean functions

We show that the problem of counting the number of n-variable unate functions reduces to the problem of counting the number of n-variable monotone functions. Using recently obtained results on n-variable monotone functions, we obtain counts of n-variable unate functions up to $n = 9$. We use an enumeration strategy to obtain the number of n-variable balanced monotone functions up to $n = 7$. We show that the problem of counting the number of n-variable balanced unate functions reduces to the problem of counting the number of n-variable balanced monotone functions, and consequently, we obtain the number of n-variable balanced unate functions up to $n = 7$. Using enumeration, we obtain the numbers of equivalence classes of n-variable balanced monotone functions, unate functions and balanced unate functions up to $n = 6$. Further, for each of the considered sub-class of n-variable monotone and unate functions, we also obtain the corresponding numbers of n-variable non-degenerate functions.

8.1 Introduction

In this chapter, it is required to compare two Boolean functions by examining their output strings. To facilitate this comparison, it is preferable to represent the Boolean functions using bit representation. It is worth mentioning that in the field of circuit theory or switching theory, unate functions are frequently studied, and the standard representation for Boolean functions in this area is the bit representation. Hence, we have chosen to adopt the same representation in this chapter. Therefore, for a positive integer n, we will consider an n-variable Boolean function f is a map $f : \{0,1\}^n \to \{0,1\}$.

For a positive integer n, an n-variable Boolean function f is a map $f : \{0,1\}^n \to \{0,1\}$. A Boolean function f is said to be monotone increasing (resp. decreasing) in the i-th variable if

$$f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \leq f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

$$(\text{resp. } f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n) \geq f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n))$$

for all possible $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \in \{0,1\}$. The function f is said to be locally monotone or unate, if for each $i \in \{1,\ldots,n\}$, it is either monotone increasing or monotone decreasing in the i-th variable. The function f is said to be monotone increasing (or, simply monotone) if for each $i \in \{1,\ldots,n\}$, it is monotone increasing in the i-th variable.

Unate functions have been studied in the literature from various viewpoints such as switching theory [105, 118, 166, 116, 17, 140], combinatorial aspects [166, 58, 84], and complexity theoretic aspects [7, 182, 121, 84]. Monotone functions have been studied much more extensively than unate functions and have many applications so much so that it is difficult to mention a few representative works.

A Boolean function is degenerate on some variable if its output does not depend on the variable, and it is said to be non-degenerate if it is not degenerate on any variable. Two Boolean functions on the same number of variables are said to be equivalent if one can be obtained from the other by a permutation of variables. The notion of equivalence partitions the set of Boolean functions into equivalence classes.

The number of n-variable monotone Boolean functions is called the n-th Dedekind number $D(n)$ after Dedekind who posed the problem in 1897. Till date, the n-th Dedekind numbers has been obtained only up to $n = 9$ (see [161, 53, 46, 172, 14, 174, 60, 90, 85]). A closed form summation formula for $D(n)$ was given in [102], though it was pointed out in [108] that using the formula to compute $D(n)$ has the same complexity as direct enumeration of all n-variable monotone Boolean functions. Dedekind numbers form the entry A000372 of [161]. The number of n-variable non-degenerate Boolean functions can be obtained as the inverse binomial transform of the Dedekind numbers and are hence also known up to $n = 9$. These numbers form the entry A006126 of [161]. The numbers of n-variable inequivalent monotone Boolean functions are known up to $n = 9$ (see [163, 136, 137]) and form the entry A003182 of [161].

A basic property of Boolean functions is balancedness. A Boolean function is said to be balanced if it takes the values 0 and 1 equal number of times. The number of n-variable balanced Boolean functions is $\binom{2^n}{2^{n-1}}$.

The focus of the present work is on counting unate and monotone Boolean functions under various restrictions. For $n \leq 5$, it is possible to enumerate all n-variable Boolean functions. Consequently, the problem of counting various sub-classes of n-variable Boolean functions becomes a reasonably simple problem. Non-triviality of counting Boolean functions

arises for $n \geq 6$.

We show that the problem of counting unate functions reduces to the problem of counting monotone functions. Since the numbers of n-variable monotone functions are known up to $n = 9$, these values immediately provide the numbers of n-variable unate functions up to $n = 9$. The problem of counting balanced monotone functions has not been considered in the literature. We use an enumeration strategy to count the number of balanced monotone functions up to $n = 7$. We show that the problem of counting balanced unate functions reduces to the problem of counting balanced monotone functions. Consequently, we obtain the numbers of n-variable balanced unate functions up to $n = 7$. We further extend these results to obtain the numbers of non-degenerate balanced monotone functions, non-degenerate unate functions, and non-degenerate balanced unate functions.

We describe a simple filtering technique for counting the number of equivalence classes of n-variable functions possessing a given property. Using this technique, we compute the number of equivalence classes of n-variable balanced monotone functions. Unlike the situation for counting functions, the problem of counting the number of equivalence classes of unate functions does not reduce to the problem of counting the number of equivalence classes of monotone functions. So to count equivalence classes of unate functions, we used a method to generate all n-variable unate functions and applied our filtering technique to obtain the number of equivalence classes of n-variable unate functions. This allowed us to obtain the numbers of equivalence classes of n-variable unate and balanced unate functions up to $n = 6$. We further extend these results to obtain the numbers of equivalence classes of n-variable non-degenerate monotone functions up to $n = 9$. Moreover, we obtain the numbers of equivalence classes of n-variable balanced monotone functions, non-degenerate balanced monotone functions, non-degenerate unate functions and non-degenerate balanced unate functions up to $n = 6$.

To summarise, the new results that we obtain for monotone and unate functions are the following.

Monotone:

1. Numbers of n-variable balanced monotone functions and n-variable non-degenerate balanced monotone functions up to $n = 7$.
2. Numbers of equivalence classes of n-variable non-degenerate monotone functions up to $n = 9$.
3. Numbers of equivalence classes of n-variable balanced monotone functions, and

n-variable non-degenerate balanced monotone functions up to $n = 6$.

Unate:

1. Numbers of n-variable unate functions and n-variable non-degenerate unate functions up to $n = 9$.
2. Numbers of n-variable balanced unate functions and n-variable non-degenerate balanced unate functions up to $n = 7$.
3. Numbers of equivalence classes of n-variable unate functions, n-variable non-degenerate unate functions, n-variable balanced unate functions, and n-variable non-degenerate balanced unate functions up to $n = 6$.

Related counts: The number of NPN-equivalence classes[1] of unate Boolean functions has been studied (see [8] and $A003183$ in [161]). A proper subclass of unate functions is the class of unate cascade functions which have been studied in [151, 112, 124]. Entry $A005612$ in [161] provides counts of unate cascade functions.

Outline of the chapter: In Section 8.2 we describe the preliminaries and prove the mathematical results required to obtain the various counts. In Section 8.3 we address the problem of counting various sub-classes of monotone and unate functions and in Section 8.4 we take up the problem of counting equivalence classes of monotone and unate functions possessing a combination of several basic properties. Finally, Section 8.5 provides the concluding remarks.

8.2 Mathematical Results

We fix some terminology and notation. The cardinality of a finite set S will be denoted as $\#S$. For $x, y \in \{0, 1\}$, xy and $x \oplus y$ denote the AND and XOR operations respectively, and $\overline{x}$ denotes the complement (or negation) of x.

Elements of $\{0, 1\}^n$, $n \geq 2$, are n-bit strings (or vectors) and will be denoted using bold font. Given $n \geq 2$ and $1 \leq i \leq n$, by $\mathbf{e}_i$ we will denote the n-bit string whose i-th bit is 1 and is 0 elsewhere.

[1]Two Boolean functions are said to be NPN equivalent, if one can be obtained from the other by some combination of the following operations: a permutation of the variables, negation of a subset of the variables, and negation of the output. We say that two functions are NPN inequivalent if they are not NPN equivalent.

Let f be an n-variable Boolean function. The weight $\mathsf{wt}(f)$ of f is the size of its support, i.e. $\mathsf{wt}(f) = \#\{\mathbf{x} : f(\mathbf{x}) = 1\}$. An n-variable Boolean function f can be uniquely represented by a binary string of length 2^n in the following manner: for $0 \leq i < 2^n$, the i-th bit of the string is the value of f on the n-bit binary representation of i. We will use the same notation f to denote the string representation of f. So $f_0 \cdots f_{2^n-1}$ is the bit string of length 2^n which represents f.

By $\overline{f}$, we will denote the negation of f, i.e. $\overline{f}(\mathbf{x}) = 1$ if and only if $f(\mathbf{x}) = 0$. Let f^r be a Boolean function defined as $f^r(x_1, \ldots, x_n) = f(\overline{x}_1, \ldots, \overline{x}_n)$. The bit string representation of f^r is the reverse of the bit string representation of f.

Let g and h be two n-variable Boolean functions having string representations $g_0 \cdots g_{2^n-1}$ and $h_0 \cdots h_{2^n-1}$. We write $g \leq h$ if $g_i \leq h_i$ for $i = 0, \ldots, 2^n - 1$. From g and h, it is possible to construct an $(n+1)$-variable function f whose string representation is obtained by concatenating the string representations of g and h. We denote this construction as $f = g||h$. For $(x_1, \ldots, x_{n+1}) \in \{0,1\}^{n+1}$, we have

$$f(x_1, \ldots, x_{n+1}) \;=\; \overline{x}_1 g(x_2, \ldots, x_{n+1}) \oplus x_1 h(x_2, \ldots, x_{n+1}). \tag{8.1}$$

An n-variable Boolean function f is said to be non-degenerate on the i-th variable, $1 \leq i \leq n$, if there is an $\boldsymbol{\alpha} \in \{0,1\}^n$ such that $f(\boldsymbol{\alpha}) \neq f(\boldsymbol{\alpha} \oplus \mathbf{e}_i)$. The function f is said to be non-degenerate, if it is non-degenerate on all the n variables.

By a property $\mathcal{P}$ of the set of all Boolean functions, we will mean a subset of the set of all Boolean functions. For example, the property $\mathcal{P}$ could be the property of being balanced, being monotone, being unate, being non-degenerate, or a combination of these properties, where a combination of properties is given by the intersection of the corresponding subsets of Boolean functions. For $n \geq 0$, let P_n denote the number of n-variable Boolean functions possessing the property $\mathcal{P}$, and let $\mathsf{nd\text{-}P}_n$ denote the number of n-variable non-degenerate Boolean functions possessing the property $\mathcal{P}$. Since an n-variable function can be non-degenerate on i variables for some $i \in \{0, \ldots, n\}$ and the i variables can be chosen from the n variables in $\binom{n}{i}$ ways, we obtain the following result which shows that the sequence $\{\mathsf{P}_n\}_{n \geq 0}$ is given by the binomial transform of the sequence $\{\mathsf{nd\text{-}P}_n\}_{n \geq 0}$.

Proposition 9 *For any property $\mathcal{P}$ of Boolean functions,*

$$\mathsf{P}_n \;=\; \sum_{i=0}^{n} \binom{n}{i} \mathsf{nd\text{-}P}_i. \tag{8.2}$$

Consequently,

$$\mathsf{nd\text{-}P}_n = \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\mathsf{P}_i. \tag{8.3}$$

Remark 13 *We assume that for $n = 0$, there are two n-variable, non-degenerate, monotone (and hence unate), and unbalanced Boolean functions whose string representations are 0 and 1.*

For $n \geq 0$, let $\mathsf{A}_n = 2^{2^n}$ be the number of all n-variable Boolean functions, and let $\mathsf{B}_n = \binom{2^n}{2^{n-1}}$ be the number of n-variable balanced Boolean functions. Let $\mathsf{nd\text{-}A}_n$ be the number of all non-degenerate n-variable Boolean functions, and $\mathsf{nd\text{-}B}_n$ be the number of all non-degenerate n-variable balanced Boolean functions. Using Proposition 9, we obtain

$$\mathsf{nd\text{-}A}_n = \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\cdot 2^{2^i} \quad \text{and} \quad \mathsf{nd\text{-}B}_n = \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\cdot\binom{2^i}{2^{i-1}}.$$

For $n \geq 0$, by $\mathsf{M}_n, \mathsf{BM}_n, \mathsf{U}_n$ and BU_n, we will denote the numbers of n-variable monotone, balanced-monotone, unate, and balanced-unate functions respectively, and by $\mathsf{nd\text{-}M}_n, \mathsf{nd\text{-}BM}_n, \mathsf{nd\text{-}U}_n$ and $\mathsf{nd\text{-}BU}_n$ we will denote the corresponding numbers of non-degenerate functions. The relations between the number of n-variable functions possessing one of these properties and the number of non-degenerate n-variable functions possessing the corresponding property are obtained from Proposition 9. Note that M_n is the n-th Dedekind number $D(n)$.

The following result relates the numbers of monotone and unate Boolean functions.

Proposition 10 *For $n \geq 0$, the following holds.*

$$\mathsf{nd\text{-}U}_n = 2^n \cdot \mathsf{nd\text{-}M}_n, \tag{8.4}$$

$$\mathsf{nd\text{-}BU}_n = 2^n \cdot \mathsf{nd\text{-}BM}_n, \tag{8.5}$$

$$\mathsf{U}_n \leq 2^n \cdot \mathsf{M}_n, \tag{8.6}$$

$$\mathsf{BU}_n \leq 2^n \cdot \mathsf{BM}_n, . \tag{8.7}$$

Proof: First we consider (8.4) and (8.5). We prove (8.4), the proof of (8.5) being similar.

Let f be an n-variable monotone Boolean function. Then it is easy to see that for any $\boldsymbol{\alpha} \in \{0,1\}^n$, the n-variable function $f_{\boldsymbol{\alpha}}$ is unate, where $f_{\boldsymbol{\alpha}}$ is defined as $f_{\boldsymbol{\alpha}}(\mathbf{x}) = f(\mathbf{x} \oplus \boldsymbol{\alpha})$ for all $\mathbf{x} \in \{0,1\}^n$. The proof of (8.4) follows from the following claim.

Claim: If f is monotone, then the 2^n possible functions $f_{\boldsymbol{\alpha}}$ corresponding to the 2^n possible $\boldsymbol{\alpha}$'s are distinct if and only if f is non-degenerate.

Proof of the claim: Suppose f is degenerate on the i-th variable. Then f and $f_{\mathbf{e}_i}$ are equal. This proves one side of the claim. So suppose that f is non-degenerate. We have to show that for $\boldsymbol{\alpha} \neq \boldsymbol{\beta}$, $f_{\boldsymbol{\alpha}}$ and $f_{\boldsymbol{\beta}}$ are distinct functions. Let if possible $f_{\boldsymbol{\alpha}}$ and $f_{\boldsymbol{\beta}}$ be equal. Note that since f is non-degenerate, both $f_{\boldsymbol{\alpha}}$ and $f_{\boldsymbol{\beta}}$ are also non-degenerate. Since $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ are distinct, there is a j in $\{1, \ldots, n\}$ such that $\alpha_j \neq \beta_j$. Suppose without loss of generality that $\alpha_j = 0$ and $\beta_j = 1$. Since f is monotone, it is monotone increasing in all variables and hence in the j-th variable. Further, since $\alpha_j = 0$, $f_{\boldsymbol{\alpha}}$ is monotone increasing in the j-th variable and since $\beta_j = 1$, $f_{\boldsymbol{\beta}}$ is monotone decreasing in the j-th variable. From $f_{\boldsymbol{\alpha}}$ is monotone increasing in the j-th variable we have that for all $\mathbf{y} = (y_1, \ldots, y_n) \in \{0,1\}^n$ with $y_j = 0$, $f_{\boldsymbol{\alpha}}(\mathbf{y}) \leq f_{\boldsymbol{\alpha}}(\mathbf{y} \oplus \mathbf{e}_j)$. Further, since $f_{\boldsymbol{\alpha}}$ is non-degenerate, and hence non-degenerate on the j-th variable, equality cannot hold everywhere, i.e. there is a $\mathbf{z} = (z_1, \ldots, z_n) \in \{0,1\}^n$ with $z_j = 0$, such that $f_{\boldsymbol{\alpha}}(\mathbf{z}) = 0$ and $f_{\boldsymbol{\alpha}}(\mathbf{z} \oplus \mathbf{e}_j) = 1$. Since $f_{\boldsymbol{\alpha}}$ and $f_{\boldsymbol{\beta}}$ are assumed to be equal, it follows that $f_{\boldsymbol{\beta}}(\mathbf{z}) = 0$ and $f_{\boldsymbol{\beta}}(\mathbf{z} \oplus \mathbf{e}_j) = 1$, which contradicts the fact that $f_{\boldsymbol{\beta}}$ is monotone decreasing in the j-th variable. This proves the claim.

Next we consider (8.6) and (8.7). We provide the proof of (8.6), the proof of (8.7) being similar. The relation given by (8.6) can be obtained from (8.4) and Proposition 9 using the following calculation.

$$\mathsf{U}_n \;=\; \sum_{i=0}^{n} \binom{n}{i} \mathsf{nd\text{-}U}_i = \sum_{i=0}^{n} \left(\binom{n}{i} \cdot 2^i \cdot \mathsf{nd\text{-}M}_i \right) \leq 2^n \cdot \sum_{i=0}^{n} \binom{n}{i} \mathsf{nd\text{-}M}_i = 2^n \cdot \mathsf{M}_n.$$

□

We record two known facts about monotone functions.

Proposition 11 *[6] Let g and h be n-variable Boolean functions and $f = g||h$. Then f is a monotone function if and only if g and h are both monotone functions and $g \leq h$.*

Proposition 12 *(A003183 of [161]) If f is a monotone function then $\overline{f^r}$ is also a monotone function.*

Next we present some results on unate and monotone functions which will be useful in our enumeration strategy. The first result is the analogue of Proposition 11 for unate functions.

Proposition 13 *Let g and h be n-variable functions and $f = g||h$. Then f is a unate function if and only if g and h are both unate functions satisfying the following two conditions.*

1. *For each variable, g and h are either both monotone increasing, or both monotone decreasing.*

2. *Either $g \leq h$ or $h \leq g$.*

Proof: First consider the proof of the "if" part. Suppose g and h are unate functions satisfying the stated condition. We have to show that for each variable, f is either monotone increasing, or monotone decreasing. Consider the variable x_1. If $g \leq h$, then from (8.1), f is monotone increasing on x_1, while if $g \geq h$, then again from (8.1), f is monotone decreasing on x_1. Now consider any variable x_i, with $i \geq 2$. If g and h are both monotone increasing on x_i, then f is also monotone increasing on x_i, while if g and h are both monotone decreasing on x_i, then f is also monotone decreasing on x_i. Since for each variable, f is either monotone increasing, or monotone decreasing, it follows that f is a unate function.

For the converse, suppose that f is a unate functions. Then for each variable x_i, $i \geq 1$, f is either monotone increasing or monotone decreasing. From (8.1), it follows that for each variable x_i, $i \geq 2$, g and h are either both monotone increasing, or both monotone decreasing. So in particular, g and h are unate. If f is monotone increasing for x_1, then $g \leq h$ and if f is monotone decreasing for x_1, then $g \geq h$. □

Proposition 14 *If f is a unate function then $\overline{f}$ is also a unate function.*

Proof: The proof is by induction on the number of variables n. The base case is $n = 1$ and is trivial. Suppose the result holds for some $n \geq 1$. Suppose that f is an $(n+1)$-variable unate function. Then f can be written as $f = g||h$, where g and h are n-variable unate functions satisfying the conditions in Proposition 13. Then $\overline{f} = \overline{g}||\overline{h}$. By induction hypothesis, $\overline{g}$ and $\overline{h}$ are n-variable unate functions and the conditions in Proposition 13 hold for $\overline{g}$ and $\overline{h}$. So $\overline{f}$ is a unate function. □

For $0 \leq w \leq 2^n$, let $\mathsf{M}_{n,w}$ (resp. $\mathsf{U}_{n,w}$) be the number of n-variable monotone (resp. unate) Boolean functions of weight w.

Proposition 15 *For any $n \geq 1$ and weight $w \in [0, 2^n]$, $\mathsf{M}_{n,w} = \mathsf{M}_{n,2^n-w}$.*

Proof: Proposition 12 sets up a one-one correspondence between n-variable monotone functions having weight w and n-variable monotone functions having weight $2^n - w$. This shows that $\mathsf{M}_{n,w} = \mathsf{M}_{n,2^n-w}$. □

Proposition 16 *For any $n \geq 1$ and weight $w \in [0, 2^n]$, $\mathsf{U}_{n,w} = \mathsf{U}_{n,2^n-w}$.*

Proof: Proposition 14 sets up a one-one correspondence between n-variable unate functions having weight w and n-variable unate functions having weight $2^n - w$. This shows that $\mathsf{U}_{n,w} = \mathsf{U}_{n,2^n-w}$. □

8.2.1 Equivalence

Two Boolean functions are equivalent if they have the same number of variables and one can be obtained from the other by a permutation of variables. Let $\mathcal{P}$ be a property of Boolean functions. The set $\mathcal{P}$ is partitioned into equivalence classes by the notion of equivalence. For $n \geq 0$, let $[P]_n$ denote the number of equivalence classes of n-variable functions possessing the property $\mathcal{P}$. Also, let $\mathsf{nd}\text{-}[P]_n$ denote the number of equivalence classes of non-degenerate n-variable functions possessing the property $\mathcal{P}$.

Remark 14 *We assume that for $n = 0$, there are two equivalence classes of n-variable, non-degenerate, monotone (and hence unate), and unbalanced Boolean functions given by* [0] *and* [1].

We have the following analogue of Proposition 9.

Proposition 17 *Let $\mathcal{P}$ be a property of Boolean functions which is closed under permutation of variables (i.e. if f is in $\mathcal{P}$ and g is obtained from f by applying a permutation to the variables, then g is also in $\mathcal{P}$). Then*

$$[P]_n = \sum_{i=0}^{n} \mathsf{nd}\text{-}[P]_i. \tag{8.8}$$

Consequently, $\mathsf{nd}\text{-}[P]_n = [P]_n - [P]_{n-1}$.

For $n \geq 0$, let $[A]_n$ denote the number of equivalence classes of n-variable Boolean functions and $[B]_n$ denote the number of equivalence classes of n-variable balanced Boolean functions. The values of $[A]_n$ and $[B]_n$ can be obtained using Polya's theory (see for example [144]). Let $\mathsf{nd}\text{-}[A]_n$ denote the number of equivalence classes of n-variable non-degenerate Boolean functions and $\mathsf{nd}\text{-}[B]_n$ denote the number of equivalence classes of n-variable non-degenerate balanced Boolean functions. Using Proposition 17, $\mathsf{nd}\text{-}[A]_n = [A]_n - [A]_{n-1}$ and $\mathsf{nd}\text{-}[B]_n = [B]_n - [B]_{n-1}$.

For $n \geq 0$, by $[M]_n$, $[BM]_n$, $[U]_n$ and $[BU]_n$ we will denote the numbers of equivalence classes of n-variable monotone, balanced-monotone, unate, and balanced-unate functions

respectively and by $\mathsf{nd}\text{-}[M]_n$, $\mathsf{nd}\text{-}[BM]_n$, $\mathsf{nd}\text{-}[U]_n$ and $\mathsf{nd}\text{-}[BU]_n$ we will denote the corresponding numbers of equivalence classes of non-degenerate functions. The following result is the analogue of Proposition 10.

Proposition 18 *For $n \geq 0$, the following holds.*

$$\mathsf{nd}\text{-}[U]_n \leq 2^n \cdot \mathsf{nd}\text{-}[M]_n, \tag{8.9}$$

$$\mathsf{nd}\text{-}[BU]_n \leq 2^n \cdot \mathsf{nd}\text{-}[BM]_n, \tag{8.10}$$

$$[U]_n \leq 2^n \cdot [M]_n, \tag{8.11}$$

$$[BU]_n \leq 2^n \cdot [BM]_n, \tag{8.12}$$

The relations given by (8.11) and (8.12) are analogues of (8.6) and (8.7) respectively. However, unlike (8.4) and (8.5), we do not have equalities in (8.9) and (8.10). The reason is that two distinct input translations of a non-degenerate monotone function can lead to two unate functions which are equivalent. An example is the following. Suppose $f(X_1, X_2) = X_1X_2$, i.e. f is the AND function. Let $g(X_1, X_2) = f(1 \oplus X_1, X_2) = (1 \oplus X_1)X_2$ and $h(X_1, X_2) = f(X_1, 1 \oplus X_2) = X_1(1 \oplus X_2)$. Then $g(X_1, X_2) = h(X_2, X_1)$, i.e. g and h are distinct, but equivalent unate functions obtained by distinct input translations from the monotone function f.

8.3 Counting Functions

In this section, we consider the problem of counting various sub-classes of monotone and unate Boolean functions.

8.3.1 Monotone Functions

Note that M_n is the n-th Dedekind number. For $0 \leq n \leq 9$, the values of M_n are known [161], with the value of M_9 being obtained recently and independently by two groups of researchers [85, 90]. The values of M_n form entry A000372 of [161]. The numbers of non-degenerate n-variable monotone functions, $\mathsf{nd}\text{-}\mathsf{M}_n$, form entry A006126 of [161].

We used enumeration to obtain the number BM_n of n-variable balanced monotone functions. For $n \leq 6$, we enumerated all monotone functions and counted only the balanced functions. Our strategy for enumerating monotone functions is based on Proposition 11.

The approach is the following. First generate all 1-variable monotone functions and store these. For $n \geq 2$, to generate all n-variable monotone functions, we consider each pair (g, h) of $(n-1)$-variable monotone functions and check whether the pair satisfies the condition of Proposition 11. If it does, then $f = g||h$ is stored. To generate all n-variable monotone functions, this approach requires considering $(\mathsf{M}_{n-1})^2$ pairs. The enumeration and filtering out unbalanced functions allows us to obtain the values of BM_n, for $n = 1, \ldots, 6$.

Remark 15 *To obtain a faster method, one may consider generating only non-degenerate functions using Proposition 11. This, however, does not work. It is indeed true that if g and h are distinct non-degenerate functions, $f = g||h$ is also non-degenerate. On the other hand, it is possible that one of g or h is degenerate, but f is non-degenerate. For example, take g to be the 2-variable constant function whose string representation is* 0000*, and h to be the 2-variable AND function whose string representation is given by* 0001*. Then the string representation of the 3-variable function $f = g||h$ is* 00000001 *which is the AND of the three variables and hence non-degenerate. So the set of all non-degenerate n-variable monotone functions cannot be obtained by concatenating only non-degenerate $(n-1)$-variable monotone functions.*

To obtain BM_7 we used a faster method. After enumerating all 6-variable monotone functions, we divided these functions into groups where all functions in the same group have the same weight. Our modified strategy is to take an n-variable monotone functions g and h, where g has weight w and h has weight $2^n - w$ and check whether $g \leq h$. If the check passes, then we generate the $(n+1)$-variable balanced monotone function $f = g||h$. Recall that for $0 \leq w \leq 2^n$, there are $\mathsf{M}_{n,w}$ n-variable monotone functions having weight w. The number of pairs needed to be considered by the modified method is

$$\sum_{w=0}^{2^n} \mathsf{M}_{n,w}\mathsf{M}_{n,2^n-w} = \sum_{w=0}^{2^n} \left(\mathsf{M}_{n,w}\right)^2,$$

where the equality follows from Proposition 15. Substituting $n = 6$ and using the values of $\mathsf{M}_{6,w}$ obtained through enumeration, we find that the modified strategy for generating 7-variable balanced monotone functions requires considering $\sum_{w=0}^{64} \left(\mathsf{M}_{6,w}\right)^2 \approx 2^{40}$ pairs, while the previous strategy would have required considering $(\mathsf{M}_6)^2 \approx 2^{45}$ pairs.

Remark 16 *Note that the above procedure to generate balanced monotone functions can be applied only once. It uses the set of all n-variable monotone functions to generate the set of all $(n+1)$-variable balanced monotone functions. Since this does not provide all $(n+1)$-*

n	BM_n	$\mathsf{nd\text{-}BM}_n$
0	0	0
1	1	1
2	2	0
3	4	1
4	24	16
5	621	526
6	492288	488866
7	81203064840	81199631130

Table 8.1: Numbers of n-variable balanced monotone and non-degenerate balanced monotone functions for $0 \leq n \leq 7$.

n	U_n	$\mathsf{nd\text{-}U}_n$
0	2	2
1	4	2
2	14	8
3	104	72
4	2170	1824
5	230540	220608
6	499596550	498243968
7	309075799150640	309072306743552
8	1436939192807139442941681 8	1436939192559880201215 1296
9	1466299277661687863684516782900411107623160 52	1466299277661687862391271509485252477296604 16

Table 8.2: Numbers of n-variable unate and non-degenerate unate functions for $0 \leq n \leq 9$.

variable monotone functions, it cannot be applied to generate the set of all $(n+2)$*-variable balanced monotone functions.*

Having obtained BM_n, for $n = 1, \ldots, 7$, we use Proposition 9 to obtain the values of $\mathsf{nd\text{-}BM}_n$, i.e. the number of n-variable non-degenerate balanced monotone functions. The obtained values of BM_n and $\mathsf{nd\text{-}BM}_n$ are given in Table 8.1.

8.3.2 Unate Functions

The problem of counting unate functions reduces to the problem of counting monotone functions in the following manner. Suppose we wish to obtain the number U_n of n-variable unate functions. Using Proposition 9, this reduces to the problem of obtaining $\mathsf{nd\text{-}U}_i$, for $0 \leq i \leq n$. From (8.4), this reduces to the problem of obtaining $\mathsf{nd\text{-}M}_i$ for $0 \leq i \leq n$. Using another application of Proposition 9 reduces the problem of obtaining $\mathsf{nd\text{-}M}_i$ to that of obtaining M_j for $0 \leq j \leq i$. So to obtain U_n, it is sufficient to know M_i for $0 \leq i \leq n$. Since the values of M_i are known for $0 \leq i \leq 9$, we can obtain the values of U_n for $0 \leq n \leq 9$. From these values, using Proposition 9, we obtain the values of $\mathsf{nd\text{-}U}_n$ for $0 \leq n \leq 9$. The values of U_n and $\mathsf{nd\text{-}U}_n$ are shown in Table 8.2.

n	BU_n	$\mathsf{nd\text{-}BU}_n$
0	0	0
1	2	2
2	4	0
3	14	8
4	296	256
5	18202	16832
6	31392428	31287424
7	10393772159334	10393552784640

Table 8.3: Numbers of n-variable balanced unate and non-degenerate balanced unate functions for $0 \leq n \leq 7$.

In a similar manner, using Proposition 9 and (8.5), the problem of counting balanced unate functions can be reduced to the problem of counting balanced monotone functions. Since we have obtained the values of BM_i for $0 \leq i \leq 7$, we obtain the values of BU_n for $0 \leq n \leq 7$. Using Proposition 9, this gives us the values of $\mathsf{nd\text{-}BU}_n$ for $0 \leq n \leq 7$. The values of BU_n and $\mathsf{nd\text{-}BU}_n$ are shown in Table 8.3.

8.4 Counting Equivalence Classes of Functions

In this section, we present the results on numbers of equivalence classes of various subsets of monotone and unate functions.

8.4.1 Filtering Procedure

The basic problem of enumerating equivalence classes is the following. Let $\mathcal{S}$ be a subset of the set of all n-variable Boolean functions. Given $\mathcal{S}$, we wish to generate a set $\mathcal{T} \subseteq \mathcal{S}$ of functions such that no two functions in $\mathcal{T}$ are equivalent, and each function in $\mathcal{S}$ is equivalent to some function in $\mathcal{T}$. The technique for such filtering is the following.

Given a permutation π of $\{1, \ldots, n\}$, we define a permutation $\pi^\star$ of $\{0, \ldots, 2^n - 1\}$ as follows. For $i \in \{0, \ldots, 2^n - 1\}$, let $(i_1, \ldots, i_n)$ be the n-bit binary representation of i. Then $\pi^\star(i) = j$, where the n-bit binary representation of j is $(j_{\pi(1)}, \ldots, j_{\pi(n)})$. Given an n-variable function f, let f^π denote the function such that for all $(x_1, \ldots, x_n) \in \{0,1\}^n$, $f^\pi(x_1, \ldots, x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$. Suppose $f_0 \cdots f_{2^n-1}$ is the bit string representation of f. Then the bit string representation of f^π is $f_{\pi^\star(0)} \cdots f_{\pi^\star(2^n-1)}$.

Note that for each permutation π, the permutation $\pi^\star$ can be pre-computed and stored as an array say $P[0, \ldots, 2^n - 1]$. Suppose the bit string representation of f is stored as an

n	nd-$[M]_n$
0	2
1	1
2	2
3	5
4	20
5	180
6	16143
7	489996795
8	1392195548399980210
9	7892046358420350391355452974102593 22

Table 8.4: Numbers of equivalence classes of n-variable non-degenerate monotone functions for $0 \leq n \leq 9$.

array $A[0, \ldots, 2^n - 1]$. Then the bit string representation of f^π is obtained as the array $B[0, \ldots, 2^n - 1]$, where $B[i] = A[P[i]]$, for $i = 0, \ldots, 2^n - 1$. So obtaining f^π becomes simply a matter of array reindexing.

Consider the set of functions $\mathcal{S}$ to be filtered is given as a list of string representations of the functions. We incrementally generate $\mathcal{T}$ as follows. The first function in $\mathcal{S}$ is moved to $\mathcal{T}$. We iterate over the other functions in $\mathcal{S}$. For a function f in $\mathcal{S}$, we generate f^π for all permutations π of $\{1, \ldots, n\}$ using the technique described above. For each such f^π, we check whether it is present in $\mathcal{T}$. If none of the f^π's are present in $\mathcal{T}$, then we append f to $\mathcal{T}$. At the end of the procedure, $\mathcal{T}$ is the desired set of functions.

The check for the presence of f^π in $\mathcal{T}$ involves a search in $\mathcal{T}$. This is done using binary search. To apply binary search on a list, it is required that the list be sorted. To ensure this, we initially ensure that $\mathcal{S}$ is sorted (either by generating it in a sorted manner, or by sorting it after generation). This ensures that at any point of time, $\mathcal{T}$ is also a sorted list, so that binary search can be applied.

8.4.2 Monotone

For $n \geq 0$, the numbers $[M]_n$ of equivalence classes of n-variable monotone functions form entry A003182 of [161]. Using Proposition 17, it is possible to find the numbers nd-$[M]_n$ of equivalence classes of n-variable monotone functions. These values are shown in Table 8.4.

For $0 \leq n \leq 6$, the numbers $[BM]_n$ of equivalence classes of n-variable balanced monotone functions are obtained by applying the filtering procedure described in Section 8.4.1 to the strategy for generating balanced monotone functions described in Section 8.3.1. Next applying Proposition 17, we obtained the numbers nd-$[BM]_n$ of equivalence classes of n-

n	$[BM]_n$	nd-$[BM]_n$
0	0	0
1	1	1
2	1	0
3	2	1
4	4	2
5	16	12
6	951	935

Table 8.5: Numbers of equivalence classes of n-variable balanced monotone and non-degenerate balanced monotone functions for $0 \leq n \leq 6$.

variable non-degenerate balanced monotone functions. The values of $[BM]_n$ and nd-$[BM]_n$ are shown in Table 8.5.

We briefly consider the computation required to obtain $[BM]_7$. From Table 8.1, $\mathsf{BM}_7 = 81203064840 \approx 2^{36.24}$. For each 7-variable balanced monotone function f, it is required to consider $7! = 5040 \approx 2^{12.29}$ functions f^π for the 7! permutations π of $\{1, \ldots, 7\}$. So a total of about $2^{48.53}$ functions would have to be considered. For each of these functions a binary search is required on the partially generated set of functions $\mathcal{T}$ and requires performing about $\log_2 \#\mathcal{T}$ comparisons. So the total number of comparisons required is somewhat above 2^{50}. This amount of computation is not presently feasible on the computing resources available to us.

8.4.3 Unate

In the case of counting functions, the problems of counting unate and balanced unate functions reduce to the problems of counting monotone and balanced monotone functions respectively. In the case of counting equivalence classes of functions, such reduction is no longer possible (using the results that we could prove). The reason is that unlike (8.4) which expresses the number of non-degenerate unate functions in terms of the number of non-degenerate monotone function, the relation (8.9) only provides an upper bound on the number of equivalence classes of non-degenerate unate functions in terms of the number of equivalence classes of non-degenerate monotone functions.

In view of the above, for counting equivalence classes of unate functions, we resorted to the technique of enumerating unate functions and then using the technique described in Section 8.4.1 to obtain the number of equivalence classes.

The technique of generating all unate functions is based on Proposition 13. Along with the string representation of a unate function, we also need to record whether the function

n	$[U]_n$	nd-$[U]_n$
0	2	2
1	4	2
2	10	6
3	34	24
4	200	166
5	3466	3266
6	829774	826308

Table 8.6: Numbers of equivalence classes of n-variable unate and non-degenerate unate functions for $0 \leq n \leq 6$.

is increasing or decreasing in each of its variables. This is recorded as the signature of the function. The special cases of the two constant functions cause some complications in the definition of the signature.

For an n-variable unate function f, we define its signature, denoted $\mathsf{sig}(f)$, to be an element of $\{0,1\}^n \cup \{\mathfrak{z}, \mathfrak{o}\}$ in the following manner. If f is the constant function 1, then $\mathsf{sig}(f) = \mathfrak{o}$, if f is the constant function 0, then $\mathsf{sig}(f) = \mathfrak{z}$; otherwise $\mathsf{sig}(f)$ is an n-bit string α, where for $i = 1, \ldots, n$, $\alpha_i = 1$ if f is monotone increasing in the variable x_i, and $\alpha_i = 0$ if f is monotone decreasing in the variable x_i. The signature $\mathsf{sig}(f)$ encodes whether f is monotone increasing or monotone decreasing on each variable. The function f is both monotone increasing and monotone decreasing in all the variables if and only if it is a constant function. The signatures of the constant functions are defined appropriately.

For enumeration, the bit string representation of the functions are used. A unate function and its signature are stored as a pair. Consider the following recursive algorithm to generate all n-variable unate functions and their signatures for $n \geq 1$. At the base step, i.e. for $n = 1$, store the four pairs of 1-variable unate functions and their signatures as $(00, \mathfrak{z})$, $(01, 1)$, $(10, 0)$ and $(11, \mathfrak{o})$. Suppose that for some $n \geq 1$, we have already generated all n-variable unate functions and their signatures. The generation of all $(n+1)$-variable unate functions and their signatures are done as follows. For any two function-signature pairs $(g, \mathsf{sig}(g))$ and $(h, \mathsf{sig}(h))$, where g and h are n-variable unate functions (which are not necessarily distinct), perform the following checks:

1. Whether at least one of $\mathsf{sig}(g)$ or $\mathsf{sig}(h)$ is equal to either $\mathfrak{z}$ or $\mathfrak{o}$ (i.e. whether at least one of g or h is a constant function).

2. $\mathsf{sig}(g) = \mathsf{sig}(h) = \alpha$, and either $g \leq h$ or $h \leq g$ holds.

If either of the checks pass, then generate $f = g||h$, and determine $\mathsf{sig}(f)$ as follows.

$$\mathsf{sig}(f) \;=\; \begin{cases} \mathfrak{z} & \text{if } \mathsf{sig}(g) = \mathsf{sig}(h) = \mathfrak{z}, \\ \mathfrak{o} & \text{if } \mathsf{sig}(g) = \mathsf{sig}(h) = \mathfrak{o}, \\ 1^{n+1} & \text{if } \mathsf{sig}(g) = \mathfrak{z},\ \mathsf{sig}(h) = \mathfrak{o}, \\ 0^{n+1} & \text{if } \mathsf{sig}(g) = \mathfrak{o},\ \mathsf{sig}(h) = \mathfrak{z}, \\ 1||\alpha & \text{if } \mathsf{sig}(g) = \mathfrak{z},\ \mathsf{sig}(h) = \alpha \in \{0,1\}^n, \\ 0||\alpha & \text{if } \mathsf{sig}(g) = \mathfrak{o},\ \mathsf{sig}(h) = \alpha \in \{0,1\}^n \\ 1||\alpha & \text{if } \mathsf{sig}(g) = \alpha \in \{0,1\}^n,\ \mathsf{sig}(h) = \mathfrak{o}, \\ 0||\alpha & \text{if } \mathsf{sig}(g) = \alpha \in \{0,1\}^n,\ \mathsf{sig}(h) = \mathfrak{z}, \\ 1||\alpha & \text{if } g \leq h,\ \mathsf{sig}(g) = \mathsf{sig}(h) = \alpha \in \{0,1\}^n, \\ 0||\alpha & \text{if } g \geq h,\ \mathsf{sig}(g) = \mathsf{sig}(h) = \alpha \in \{0,1\}^n. \end{cases} \tag{8.13}$$

Store $(f, \mathsf{sig}(f))$. Proposition 13 assures us that this recursive procedure generates all $(n+1)$-variable unate functions and their signatures.

To generate all $(n+1)$-variable unate functions, the above method requires considering all pairs of n-variable unate functions, i.e. a total of $(U(n))^2$ options. Applying the fitering strategy of Section 8.4.1 we obtain the value of $[U]_n$. Next using Proposition 17 we obtain the value of $\mathsf{nd}\text{-}[U]_n$. We could perform this computation for $n \leq 6$. The obtained values of $[U]_n$ and $\mathsf{nd}\text{-}[U]_n$ are shown in Table 8.6. To generate all 7-variable unate functions using this option requires considering $(U(6))^2 \approx 2^{57.8}$ pairs of functions. This is not feasible on the computing facility available to us.

To obtain the set of n-variable balanced unate functions, after generating the set of all n-variable unate functions, we remove the ones that are unbalanced. Then to the resulting set, we apply the technique of Section 8.4.1 to obtain the number $[BU]_n$ of equivalence classes of n-variable balanced unate functions. Subsequently, we apply Proposition 17 to obtain the number $\mathsf{nd}\text{-}[BU]_n$ of equivalence classes of n-variable non-degenerate balanced unate functions. The values of $[BU]_n$ and $\mathsf{nd}\text{-}[BU]_n$ are shown in Table 8.7

8.5 Concluding Remarks

We have obtained the numbers of n-variable unate and monotone functions possessing a combination of some basic properties. Also, we have obtained the numbers of equivalence classes of n-variable unate and monotone functions also possessing a combination of those

n	$[BU]_n$	nd-$[BU]_n$
0	0	0
1	2	2
2	2	0
3	6	4
4	24	18
5	254	230
6	50172	49918

Table 8.7: Numbers of equivalence classes of n-variable balanced unate and non-degenerate balanced unate functions for $0 \leq n \leq 6$.

same properties. Our work raises a number of questions that may be pursued in the future. One such question is whether the techniques for counting monotone functions from the recent works [90, 85] can be applied to the problem of counting balanced monotone functions. Another similar question is whether the techniques for counting the number of equivalence classes of monotone functions from [136, 137] can be applied to the problem of counting the number of equivalence classes of balanced monotone functions. A third question is whether the techniques for counting the number of equivalence classes of monotone functions from [136, 137] can be applied to the problem of counting the number of equivalence classes of unate functions. Positive answers to these questions will allow extending the results that we could obtain up to $n = 6$ or $n = 7$ to $n = 9$.

Chapter 9

Conclusion and future works

In this thesis, we investigated various aspects of Boolean functions and their properties, with a focus on the concept of influence. We began by utilizing total influence to distinguish classes of Boolean functions commonly studied in coding theory and cryptography from those in combinatorics and complexity theory.

We then introduced a novel definition of influence based on the auto-correlation function and developed a comprehensive theory around this notion. We generalized well-known results on the influence of a single variable and also obtained new characterizations of resilient and bent functions in terms of influence. Additionally, we highlighted the relationship between our introduced measure of influence and a previously proposed measure.

Furthermore, our research delved into obtaining lower bounds on the universal constant of the FMEI conjecture. We presented several construction methods for Boolean functions that provide the current best-known lower bound. Additionally, we provided a comprehensive exposition of a counterexample to the Majority is Least Stable conjecture.

Lastly, we explored unate and monotone Boolean functions, their interrelationships, and their enumeration. We found that counting unate functions can be reduced to counting monotone functions, extending our calculations to $n = 9$. Additionally, we determined counts for balanced monotone functions and unate functions up to $n = 7$. We also analyzed equivalence classes for various function types up to $n = 6$, offering insights into their structural diversity.

Future wroks: Based on the findings and open questions raised in this thesis, several promising directions for future research emerge.

1. Generalization of the edge-isoperimetric inequality: The Poincaré inequality can be viewed as an edge-isoperimetric inequality for the Boolean cube [128]. In the literature, other stronger edge-isoperimetric inequalities have been proposed by Harper [81], Khan et al. [92], and Talagrand [165]. Similar to the generalization of the Poincaré inequality (see Corollary 1), it would be worthwhile to explore generalizations for these other edge-isoperimetric inequalities.

2. Resolving open problems: Throughout this thesis, several open problems have been identified, such as settling the FEI conjecture for rotation symmetric Boolean functions and obtaining better lower bounds on the constant of the FMEI conjecture. Future research can focus on finding innovative techniques and approaches to address these problems, potentially leading to breakthroughs in understanding the structure of Boolean functions. A potential research direction could involve introducing a new method that shows promise in solving the more general Majority is Least Stable conjecture. This conjecture states that the ρ-correlated noise stability of any LTF is at least $(2/\pi)\arcsin(\rho)$ [127]. It is important to note that such an inequality is not implied by the example presented in Chapter 7. In fact, the result in the last section of Chapter 7 suggests that.

3. Finally our work in the last chapter prompts important questions for future research. Can recent counting techniques for monotone functions [90, 85] be applied to balanced monotone functions? Is it possible to adapt methods of counting equivalence classes of monotone functions to count equivalence classes of balanced monotone functions and unate functions? Answering these questions could extend our findings to $n = 9$, enriching our understanding of these Boolean function classes.

In conclusion, this thesis aspires to contribute to the field of Boolean functions by expanding our knowledge, offering fresh perspectives, and identifying open problems for further exploration. We hope that our research will inspire and guide future investigations, leading to incremental progress and advancements in this intriguing area of study.

Bibliography

[1] Berry-Esseen theorem, https://en.wikipedia.org/wiki/Berry%E2%80%93Esseen_theorem#CITEREFZolotarev1967, Accessed on October 18, 2023.

[2] Miklós Ajtai and Nathal Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[3] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[4] Kenneth J Arrow. A difficulty in the concept of social welfare. *Journal of political economy*, 58(4):328–346, 1950.

[5] Srinivasan Arunachalam, Sourav Chakraborty, Michal Koucký, Nitin Saurabh, and Ronald de Wolf. Improved bounds on Fourier entropy and min-entropy. *ACM Trans. Comput. Theory*, 13(4):22:1–22:40, 2021.

[6] Valentin Bakoev. Generating and identification of monotone Boolean functions. In *Mathematics and Education in Mathematics, Sofia*, pages 226–232, 2003.

[7] József Balogh, Dingding Dong, Bernard Lidickỳ, Nitya Mani, and Yufei Zhao. Nearly all k-SAT functions are unate. https://arxiv.org/pdf/2209.04894.pdf, 2022.

[8] Charles R. Baugh. Generation of representative functions of the NPN equivalence classes of unate Boolean functions. *IEEE Transactions on Computers*, 21(12), 1972.

[9] William Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, 102(1):159–182, 1975.

[10] Goldreich Oded Bellare, Mihir and Madhu Sudan. Free bits, pcps, and nonapproximability – towards tight results. *Electronic Colloquium on Computational Complexity*, TR95-024, 1995.

[11] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 432–441. IEEE Computer Society, 1995.

[12] Michael Ben-Or and Nathan Linial. Collective coin flipping. *Adv. Comput. Res.*, 5:91–115, 1989.

[13] Itai Benjamini, Gil Kalai, and Oded Schramm. Noise sensitivity of Boolean functions and applications to percolation. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques*, 90(1):5–43, 1999.

[14] Joel Berman and Peter Köhler. Cardinalities of finite distributive lattices. *Mitt. Math. Sem. Giessen*, 121:103–124, 1976.

[15] Anna Bernasconi, B. Codenottl, and Jeffrey VanderKam. A characterization of bent functions in terms of strongly regular graphs. *Computers, IEEE Transactions on*, 50:984 – 985, 10 2001.

[16] Andrew C. Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the American Mathematical Society*, 49:122–136, 1941. PDF.

[17] Rodolfo Betancourt. Derivation of minimum test sets for unate logical circuits. *IEEE Transactions on Computers*, 100(11):1264–1269, 1971.

[18] Aniruddha Biswas and Palash Sarkar. Separation results for Boolean function classes. *Cryptogr. Commun.*, 13(3):451–458, 2021.

[19] Aniruddha Biswas and Palash Sarkar. A lower bound on the constant in the Fourier min-entropy/influence conjecture. *Electron. Colloquium Comput. Complex.*, TR22-180, 2022.

[20] Aniruddha Biswas and Palash Sarkar. Counting unate and balanced monotone Boolean functions. *arXiv*, `https://doi.org/10.48550/arXiv.2304.14069`, 2023.

[21] Aniruddha Biswas and Palash Sarkar. Influence of a set of variables on a Boolean function. *SIAM Journal on Discrete Mathematics*, 37(3):10.1137/22M1503531, 2023.

[22] Aniruddha Biswas and Palash Sarkar. On the "majority is least stable" conjecture. *Inf. Process. Lett.*, 179:106295, 2023.

[23] Eric Blais. Testing juntas nearly optimally. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 151–158. ACM, 2009.

[24] Avrim Blum, Lisa Hellerstein, and Nick Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.

[25] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 73–83. ACM, 1990.

[26] Aline Bonami. Étude des coefficients de Fourier des fonctions de $L^p(g)$. In *Annales de l'institut Fourier*, volume 20, pages 335–402, 1970.

[27] Ravi B Boppana. The average sensitivity of bounded-depth circuits. *Information processing letters*, 63(5):257–261, 1997.

[28] Jean Bourgain, Jeff Kahn, Gil Kalai, Yitzhak Katznelson, and Nathan Linial. The influence of variables in product spaces. *Israel Journal of Mathematics*, 77:55–64, 1992.

[29] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Classification and regression trees. 1984.

[30] Jehoshua Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal on Discrete Mathematics*, 3(2):168–177, 1990.

[31] Nader H. Bshouty, Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning DNF from random walks. *Journal of Computer and System Sciences*, 71(3):250–265, 2005.

[32] Paul Camion, Claude Carlet, Pascale Charpin, and Nicolas Sendrier. On correlation-immune functions. In *Advances in Cryptology—CRYPTO'91: Proceedings 11*, pages 86–100. Springer, 1992.

[33] Anne Canteaut, Claude Carlet, Pascale Charpin, and Caroline Fontaine. Propagation characteristics and correlation-immunity of highly nonlinear Boolean functions. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 507–522. Springer, 2000.

[34] Claude Carlet. Boolean functions for cryptography and coding theory. `https://www.math.univ-paris13.fr/~carlet/book-fcts-Bool-vect-crypt-codes.pdf`.

[35] Claude Carlet. On the nonlinearity of monotone Boolean functions. *Cryptogr. Commun.*, 10(6):1051–1061, 2018.

[36] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, January 2021.

[37] Claude Carlet, David Joyner, Pantelimon Stănică, and Deng Tang. Cryptographic properties of monotone Boolean functions. *Journal of Mathematical Cryptology*, 10(1):1–14, 2016.

[38] Francis N. Castro, Luis A. Medina, and Pantelimon Stanica. Generalized walsh transforms of symmetric and rotation symmetric Boolean functions are linear recurrent. *Appl. Algebra Eng. Commun. Comput.*, 29(5):433–453, 2018.

[39] Charles Celerier, David Joyner, Caroline Melles, David Phillips, et al. On the Walsh-Hadamard transform of monotone Boolean functions. *Tbilisi Mathematical Journal*, 5(2):19–35, 2012.

[40] Sourav Chakraborty, Sushrut Karmalkar, Srijita Kundu, Satyanarayana V Lokam, and Nitin Saurabh. Fourier entropy-influence conjecture for random linear threshold functions. In *Latin American Symposium on Theoretical Informatics*, pages 275–289. Springer, 2018.

[41] Sourav Chakraborty, Sushrut Karmalkar, Srijita Kundu, Satyanarayana V. Lokam, and Nitin Saurabh. Fourier entropy-influence conjecture for random linear threshold functions. In Michael A. Bender, Martin Farach-Colton, and Miguel A. Mosteiro, editors, *LATIN 2018: Theoretical Informatics - 13th Latin American Symposium, Buenos Aires, Argentina, April 16-19, 2018, Proceedings*, volume 10807 of *Lecture Notes in Computer Science*, pages 275–289. Springer, 2018.

[42] Sourav Chakraborty, Raghav Kulkarni, Satyanarayana V Lokam, and Nitin Saurabh. Upper bounds on Fourier entropy. *Theoretical Computer Science*, 654:92–112, 2016.

[43] Seongtaek Chee, Sangjin Lee, Daiki Lee, and Soo Hak Sung. On the correlation immune functions and their nonlinearity. In *Advances in Cryptology—ASIACRYPT'96: International Conference on the Theory and Applications of Cryptology and Information Security Kyongju, Korea, November 3–7, 1996 Proceedings*, pages 232–243. Springer, 1996.

[44] Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.

[45] Chao-Kong Chow. On the characterization of threshold functions. In *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, pages 34–38. IEEE, 1961.

[46] Randolph Church. Nunmerical analysis of certain free distributive structures. *Duke Mathematical Journal*, 6(3):732 – 734, 1940.

[47] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[48] Nicolas T Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*, pages 345–359. Springer, 2003.

[49] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[50] Thomas W Cusick and Pantelimon Stanica. *Cryptographic Boolean functions and applications*. Academic Press, 2017.

[51] Bireswar Das, Manjish Pal, and Vijay Visavaliya. The entropy influence conjecture revisited. *arXiv preprint arXiv:1110.4301*, 2011.

[52] Ronald de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory Comput.*, 1:1–20, 2008.

[53] Richard Dedekind. Über zerlegungen von zahlen durch ihre grössten gemeinsamen teiler. *Gesammelte Werke*, 2:103 – 148, 1897.

[54] Ilias Diakonikolas, Prasad Raghavendra, Rocco A Servedio, and Li-Yang Tan. Average sensitivity and noise sensitivity of polynomial threshold functions. *SIAM Journal on Computing*, 43(1):231–253, 2014.

[55] Irit Dinur. The pcp theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12–es, 2007.

[56] P. Erdös and A. Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[57] Carl-Gustav Esseen. *On the Liapounoff Limit of Error in the Theory of Probability*, volume 19 of *Arkiv för matematik, astronomi och fysik*. Stockholm Almqvist & Wiksell, 1942.

[58] Aaron Feigelson and Lisa Hellerstein. The forbidden projections of unate functions. *Discrete Applied Mathematics*, 77(3):221–236, 1997.

[59] W. Feller. *An Introduction to Probability Theory and Its Applications: Volume I*. Wiley series in probability and mathematical statistics. John Wiley & Sons, 1968.

[60] Robert Fidytek, Andrzej W. Mostowski, Rafal Somla, and Andrzej Szepietowski. Algorithms counting monotone Boolean functions. *Inf. Process. Lett.*, 79(5):203–209, 2001.

[61] Yuval Filmus, Hamed Hatami, Steven Heilman, Elchanan Mossel, Ryan O'Donnell, Sushant Sachdeva, Andrew Wan, and Karl Wimmer. Real analysis in computer science: A collection of open problems (2014). *Preprint available at*.

[62] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 103–112. IEEE Computer Society, 2002.

[63] Robert Fortet. Applications de l'algebre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4(14):17–26, 1960.

[64] Robert Fortet. L'algebre de boole et ses applications en recherche opérationnelle. *Trabajos de Estadistica*, 11:111–118, 1960.

[65] Ehud Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–35, 1998.

[66] Ehud Friedgut and Gil Kalai. Every monotone graph property has a sharp threshold. *Proceedings of the American mathematical Society*, 124(10):2993–3002, 1996.

[67] Ehud Friedgut, Gil Kalai, and Assaf Naor. Boolean functions whose Fourier transform is concentrated on the first two levels. *Advances in Applied Mathematics*, 29(3):427–437, 2002.

[68] Sugata Gangopadhyay and Pantelimon Stanica. The Fourier entropy-influence conjecture holds for a log-density 1 class of cryptographic Boolean functions. *IACR Cryptol. ePrint Arch.*, 2014:54, 2014.

[69] Sugata Gangopadhyay and Pantelimon Stănică. The Fourier entropy-influence conjecture holds for a log-density 1 class of cryptographic Boolean functions. Cryptology ePrint Archive, Paper 2014/054, 2014. `https://eprint.iacr.org/2014/054`.

[70] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

[71] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[72] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989.

[73] Craig Gotsman and Nathan Linial. Spectral properties of threshold functions. *Combinatorica*, 14(1):35–50, 1994.

[74] Kishan Chand Gupta. *Cryptographic and combinatorial properties of Boolean functions and s-boxes*. PhD thesis, Indian Statistical Institute-Kolkata, 2004.

[75] Kishan Chand Gupta and Palash Sarkar. Improved construction of nonlinear resilient s-boxes. In *ASIACRYPT*, pages 466–483. Springer, 2002.

[76] Kishan Chand Gupta and Palash Sarkar. Construction of perfect nonlinear and maximally nonlinear multiple-output Boolean functions satisfying higher order strict avalanche criteria. *IEEE transactions on information theory*, 50(11):2886–2893, 2004.

[77] Kishan Chand Gupta and Palash Sarkar. Construction of high degree resilient s-boxes with improved nonlinearity. *Information processing letters*, 95(3):413–417, 2005.

[78] Kishan Chand Gupta and Palash Sarkar. Toward a general correlation theorem. *IEEE Trans. Inf. Theory*, 51(9):3297–3302, 2005.

[79] Carl gustav Esseen. A moment inequality with an application to the central limit theorem. *Scandinavian Actuarial Journal*, 1956:160–170, 1956.

[80] P.L. Hammer and S. Rudeanu. *Boolean Methods in Operations Research and Related Areas*. Econometrics and operations research. Springer-Verlag, 1968.

[81] Lawrence Hueston Harper. Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics*, 12(1):131–135, 1964.

[82] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[83] Johan Torkel Håstad. *Computational limitations for small-depth circuits*. MIT press, 1987.

[84] Yutaka Hata, Masaharu Yuhara, Fujio Miyawaki, and Kazuharu Yamato. On the complexity of enumerations for multiple-valued Kleenean functions and unate functions. In *1991 Proceedings of the Twenty-First International Symposium on Multiple-Valued Logic*, pages 55–56. IEEE Computer Society, 1991.

[85] Lennart Van Hirtum, Patrick De Causmaecker, Jens Goemaere, Tobias Kenter, Heinrich Riebler, Michael Lass, and Christian Plessl. A computation of $D(9)$ using FPGA supercomputing. `https://arxiv.org/abs/2304.03039`, 2023.

[86] Rani Hod. Improved lower bounds for the Fourier entropy/influence conjecture via lexicographic functions. *arXiv preprint arXiv:1711.00762*, 2017.

[87] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[88] Jeffrey C Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

[89] Vishesh Jain. A counterexample to the "Majority is Least Stable" conjecture. *arXiv preprint arXiv:1703.07657*, 2017.

[90] Christian Jäkel. A computation of the ninth Dedekind number. `https://arxiv.org/abs/2304.00895`, 2023.

[91] Thomas Johansson and Enes Pasalic. A construction of resilient functions with high nonlinearity. *IEEE Transactions on Information Theory*, 49(2):494–501, 2003.

[92] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 68–80. IEEE Computer Society, 1988.

[93] Gil Kalai. A Fourier-theoretic perspective on the condorcet paradox and arrow's theorem. *Advances in Applied Mathematics*, 29(3):412–426, 2002.

[94] Gil Kalai. Boolean functions: Influence, threshold and noise. In *European Congress of Mathematics (2016)*, pages 85–110. European Mathematical Society, 2018.

[95] Gil Kalai and Shmuel Safra. Threshold phenomena and influence with some perspectives from mathematics. *Computer Science and Economics*, 2004.

[96] Michael Kearns and Yishay Mansour. On the boosting ability of top-down decision tree learning algorithms. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 459–468, 1996.

[97] Anthony M Kerdock. A class of low-rate nonlinear binary codes. *Information and control*, 20(2):182–187, 1972.

[98] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002.

[99] Subhash Khot. Inapproximability of np-complete problems, discrete Fourier analysis, and geometry. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 2676–2697. World Scientific, 2010.

[100] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[101] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2- ε. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[102] Andrzej Kisielewicz. A solution of Dedekind's problem on the number of isotone Boolean functions. *Journal fur die Reine und Angewandte Mathematik*, 1988(386):139 – 144, 1988.

[103] Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004.

[104] Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{O(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004.

[105] Zvi Kohavi. *Switching and finite automata theory*. McGraw-Hill (New York, NY [ua]), 1970.

[106] V. Korolev and Irina Shevtsova. On the upper bound for the absolute constant in the Berry–Esseen inequality. *Theory of Probability and its Applications*, 54, 01 2010.

[107] V. Yu. Korolev and I. G. Shevtsova. On the upper bound for the absolute constant in the Berry–Esseen inequality. *Theory of Probability & Its Applications*, 54(4):638–658, 2010.

[108] Aleksej D. Korshunov. Monotone Boolean functions. *Russian Mathematical Surveys*, 58(5):929 – 1001, 2003.

[109] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 455–464, 1991.

[110] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

[111] Florence J. MacWilliams and Neil J. A. Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[112] Karuna K Maitra. Cascaded switching networks of two-input flexible cells. *IRE Transactions on Electronic Computers*, (2):136–143, 1962.

[113] Subhamoy Maitra and Palash Sarkar. Characterization of symmetric bent functions – an elementary proof. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 43:227–230, 2002.

[114] Yishay Mansour. An $o(n \log \log n)$ learning algorithm for DNF under the uniform distribution. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 53–61, 1992.

[115] Grigorii Aleksandrovich Margulis. Probabilistic characteristics of graphs with large connectivity. *Problemy peredachi informatsii*, 10(2):101–108, 1974.

[116] William S. Matheson. Recognition of monotonic and unate cascade realizable functions using an informational model of switching circuits. *IEEE Transactions on Computers*, 100(10):1214–1219, 1971.

[117] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Advances in Cryptology—EUROCRYPT'93: Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings 12*, pages 386–397. Springer, 1994.

[118] Robert McNaughton. Unate truth functions. *IRE Transactions on Electronic Computers*, EC-10(1):1–6, 1961.

[119] Pierrick Méaux. On the fast algebraic immunity of majority functions. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, volume 11774 of *Lecture Notes in Computer Science*, pages 86–105. Springer, 2019.

[120] Willi Meier and Othmar Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of cryptology*, 1:159–176, 1989.

[121] Hiroki Morizumi. Sensitivity, block sensitivity, and certificate complexity of unate functions and read-once functions. In *IFIP International Conference on Theoretical Computer Science*, pages 104–110. Springer, 2014.

[122] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. volume 171, pages 295–341. Annals of Mathematics, 2010.

[123] Elchanan Mossel, Ryan O'Donnell, and Rocco A Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004.

[124] Amar Mukhopadhyay. Unate cellular logic. *IEEE Transactions on Computers*, 100(2):114–121, 1969.

[125] David E Muller. Application of Boolean algebra to switching circuit design and to error detection. *Transactions of the IRE professional group on electronic computers*, (3):6–12, 1954.

[126] Kaisa Nyberg. Perfect nonlinear s-boxes. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*, pages 378–386. Springer, 1991.

[127] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[128] Ryan O'Donnell. Analysis of Boolean functions, 2021.

[129] Ryan O'Donnell and Rocco A Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007.

[130] Ryan O'Donnell and Li-Yang Tan. A composition theorem for the Fourier entropy-influence conjecture. In *International Colloquium on Automata, Languages, and Programming*, pages 780–791. Springer, 2013.

[131] Ryan O'Donnell, John Wright, and Yuan Zhou. The Fourier entropy–influence conjecture for certain classes of Boolean functions. In *International Colloquium on Automata, Languages, and Programming*, pages 330–341. Springer, 2011.

[132] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Proclaiming dictators and juntas or testing Boolean formulae. In *RANDOM 2001 Berkeley, CA, USA, August 18-20, 2001, Proceedings*, volume 2129 of *Lecture Notes in Computer Science*, pages 273–284. Springer, 2001.

[133] Enes Pasalic and Subhamoy Maitra. Linear codes in generalized construction of resilient functions with very high nonlinearity. *IEEE Transactions on Information Theory*, 48(8):2182–2191, 2002.

[134] Nick J. Patterson and Douglas H. Wiedemann. The covering radius of the (2^{15}, 16) reed-muller code is at least 16276. *IEEE Trans. Inf. Theory*, 29(3):354–355, 1983.

[135] Nick J. Patterson and Douglas H. Wiedemann. Correction to 'the covering radius of the (2^{15}, 16) reed-muller code is at least 16276' (may 83 354-356). *IEEE Trans. Inf. Theory*, 36(2):443, 1990.

[136] Bartłomiej Pawelski. On the number of inequivalent monotone Boolean functions of 8 variables. `https://arxiv.org/abs/2108.13997`, 2021.

[137] Bartłomiej Pawelski. On the number of inequivalent monotone Boolean functions of 9 variables. `https://arxiv.org/abs/2305.06346`, 2023.

[138] Bartłomiej Pawelski. On the number of inequivalent monotone Boolean functions of 8 variables. `https://arxiv.org/pdf/2108.13997.pdf`, 2021.

[139] Yuval Peres. Noise stability of weighted majority. In Maria Eulália Vares, Roberto Fernández, Luiz Renato Fontes, and Charles M. Newman, editors, *In and Out of Equilibrium 3: Celebrating Vladas Sidoravicius*, pages 677–682. Springer International Publishing, Cham, 2021.

[140] Vijay Pitchumani and Satish S. Soman. Functional test generation based on unate function theory. *IEEE transactions on computers*, 37(6):756–760, 1988.

[141] Bart Preneel. *Analysis and design of cryptographic hash functions*. PhD thesis, Katholieke Universiteit te Leuven Leuven, 1993.

[142] Bart Preneel, Werner Van Leekwijck, Luc Van Linden, René Govaerts, and Joos Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of of Cryptographic Techniques*, volume 473, pages 161–173. Springer, 1990.

[143] I. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, 1954.

[144] Fred Roberts and Barry Tesman. *Applied Combinatorics, 2nd edition*. Chapman and Hall/CRC, 2009.

[145] Oscar S Rothaus. On "bent" functions. *Journal of Combinatorial Theory, Series A*, 20(3):300–305, 1976.

[146] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[147] Lucio Russo. On the critical percolation probabilities. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 56(2):229–237, 1981.

[148] Steven L Salzberg. C4. 5: Programs for machine learning by John R. Quinlan. Morgan Kaufmann publishers, inc., 1993, 1994.

[149] Palash Sarkar and Subhamoy Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *EUROCRYPT*, volume 1807, pages 485–506, 2000.

[150] Palash Sarkar and Subhamoy Maitra. Construction of nonlinear resilient Boolean functions using "small" affine functions. *IEEE Trans. Inf. Theory*, 50(9):2185–2193, 2004.

[151] Tsutomu Sasao and Kozo Kinoshita. On the number of fanout-free functions and unate cascade functions. *IEEE Transactions on Computers*, 28(1):66–72, 1979.

[152] Peter Savický. On the bent Boolean functions that are symmetric. *European Journal of Combinatorics*, 15(4):407–410, 1994.

[153] Jennifer Seberry, Xian-Mo Zhang, and Yuliang Zheng. On constructions and nonlinearity of correlation immune functions. In *Advances in Cryptology—EUROCRYPT'93: Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings*, pages 181–199. Springer, 2001.

[154] Guy Shalev. On the Fourier Entropy Influence conjecture for extremal classes. *arXiv preprint arXiv:1806.03646*, 2018.

[155] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

[156] I. G. Shevtsova. Sharpening of the upper bound of the absolute constant in the Berry–Esseen inequality. *Theory of Probability & Its Applications*, 51(3):549–553, 2007.

[157] Irina Shevtsova. On the absolute constants in the berry-esseen type inequalities for identically distributed summands. `https://arxiv.org/abs/1111.6554`, 2011.

[158] I. S. Shiganov. Refinement of the upper bound of the constant in the central limit theorem. *Journal of Soviet Mathematics*, 35(3):2545–2550, 1986.

[159] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.). *IEEE Transactions on Information theory*, 30(5):776–780, 1984.

[160] Thomas Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on computers*, 34(01):81–85, 1985.

[161] N.J.A. Sloance. The online encyclopedia of integer sequences. 2011.

[162] Pantelimon Stanica and Subhamoy Maitra. A constructive count of rotation symmetric functions. *Information Processing Letters*, 88(6):299–304, 2003.

[163] Tamon Stephen and Timothy Yusun. Counting inequivalent monotone Boolean functions. *Discrete Applied Mathematics*, 167:15–24, 2014.

[164] Avishay Tal. Tight bounds on the Fourier spectrum of AC^0. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 15:1–15:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[165] Michel Talagrand. On russo's approximate zero-one law. *The Annals of Probability*, pages 1576–1587, 1994.

[166] André Thayse and Jean P. Deschamps. Logic properties of unate and of symmetric discrete functions. In *Proceedings of the sixth international symposium on Multiple-valued logic*, pages 79–87, 1976.

[167] I. S. Tyurin. On the accuracy of the Gaussian approximation. *Doklady Mathematics*, 80(3):840–843, 2009.

[168] Il'ya S Tyurin. Refinement of the upper bounds of the constants in lyapunov's theorem. *Russian Mathematical Surveys*, 65(3):586, sep 2010.

[169] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[170] Paul van Beek. An application of fourier methods to the problem of sharpening the Berry-Esseen inequality. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 23(3):187–196, 1972.

[171] Andrew Wan, John Wright, and Chenggang Wu. Decision trees, protocols and the entropy-influence conjecture. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 67–80, 2014.

[172] Morgan Ward. Note on the order of free distributive lattices. *Bull. Amer. Math. Soc.*, 52:423, 1946.

[173] A. F. Webster and S. E. Tavares. On the design of S-boxes. In *Conference on the theory and application of cryptographic techniques*, pages 523–534. Springer, 1985.

[174] Doug Wiedemann. A computation of the eighth Dedekind number. *Order*, 8(1):5 – 6, 1991.

[175] Chuan-Kun Wu and Dengguo Feng. *Boolean functions and their applications in cryptography*. Springer, 2016.

[176] Guo-Zhen Xiao and James L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Trans. Inf. Theory*, 34(3):569–571, 1988.

[177] Shengyu Zhang. Note on the average sensitivity of monotone Boolean functions. *Preprint*, page 4, 2011.

[178] Xian-Mo Zhang and Yuliang Zheng. Cryptographically resilient functions. *IEEE Transactions on Information Theory*, 43(5):1740–1747, 1997.

[179] Yuliang Zheng and Xian-Mo Zhang. Plateaued functions. In Vijay Varadharajan and Yi Mu, editors, *Information and Communication Security*, pages 284–300, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[180] Yuliang Zheng and Xian-Mo Zhang. Relationships between bent functions and complementary plateaued functions. In *International Conference on Information Security and Cryptology*, pages 60–75. Springer, 1999.

[181] V. M. Zolotarev. A sharpening of the inequality of berry-esseen. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 8(4):332–342, 1967.

[182] Uri Zwick. A $4n$ lower bound on the combinational complexity of certain symmetric boolean functions over the basis of unate dyadic boolean functions. *SIAM Journal on Computing*, 20(3):499–505, 1991.