Analytic Methods in Concrete Complexity

Li-Yang Tan

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2014
ABSTRACT

Analytic Methods in Concrete Complexity

Li-Yang Tan

This thesis studies computational complexity in concrete models of computation. We draw on a range of mathematical tools to understand the structure of Boolean functions, with analytic methods — Fourier analysis, probability theory, and approximation theory — playing a central role. These structural theorems are leveraged to obtain new computational results, both algorithmic upper bounds and complexity-theoretic lower bounds, in property testing, learning theory, and circuit complexity.

• We establish the best-known upper and lower bounds on the classical problem of testing whether an unknown Boolean function is monotone. We prove an \( \tilde{\Omega}(n^{1/5}) \) lower bound on the query complexity of non-adaptive testers, an exponential improvement over the previous lower bound of \( \Omega(\log n) \) from 2002. We complement this with an \( \tilde{O}(n^{5/6}) \)-query non-adaptive algorithm for the problem.

• We characterize the statistical query complexity of agnostically learning Boolean functions with respect to product distributions. We show that \( \ell_1 \)-approximability by low-degree polynomials, known to be sufficient for efficient learning in this setting, is in fact necessary. As an application we establish an optimal lower bound showing that no statistical query algorithm can efficiently agnostically learn monotone \( k \)-juntas for any \( k = \omega(1) \) and any constant error less than \( 1/2 \).

• We initiate a systematic study of the tradeoffs between accuracy and efficiency in Boolean circuit complexity, focusing on disjunctive normal form formulas, among the most basic types of circuits. A conceptual message that emerges is that the landscape of circuit complexity changes dramatically, both qualitatively and quantitatively, when the formula is only required to approximate a function rather than compute it exactly.
Finally we consider the Fourier Entropy-Influence Conjecture, a longstanding open problem in the analysis of Boolean functions with significant applications in learning theory, the theory of pseudorandomness, and random graph theory. We prove a composition theorem for the conjecture, broadly expanding the class of functions for which the conjecture is known to be true.
Table of Contents

List of Tables iii

1 Introduction 1
    1.1 Concrete complexity ........................................ 1
    1.2 Boolean functions ........................................... 3
    1.3 Analytic methods ............................................ 4
    1.4 Outline of this thesis ....................................... 5

I Property Testing 8

2 New Algorithms and Lower Bounds for Testing Monotonicity 9
    2.1 Background and context ....................................... 9
    2.2 A polynomial lower bound for testing monotonicity .......... 18
    2.3 Multidimensional Berry–Esséen via the Valiant–Valiant CLT  21
    2.4 General hypergrid domains ................................... 24
    2.5 An improved algorithm for testing monotonicity ............. 29

II Computational Learning Theory 39

3 Approximate Resilience and the Complexity of Agnostic Learning 40
    3.1 Background and context ....................................... 40
    3.2 Characterizing the complexity of agnostic learning .......... 51
    3.3 Monotonicity and approximate resilience ..................... 56
III Boolean Function Complexity

4 Approximating Boolean Functions by Small-Depth Circuits

5 On DNF Approximators for Monotone Boolean Functions

6 A Composition Theorem for the Fourier Entropy-Influence Conjecture

7 Conclusions

Bibliography
List of Tables

4.1 Approximating any Boolean function to constant accuracy. ............... 82
4.2 Approximating PAR\(_\text{m} \) to accuracy \( \varepsilon \). ....................... 83
Acknowledgments

I am unbelievably lucky to have had Rocco Servedio as my advisor. I cannot imagine a more meaningful and enjoyable Ph.D. experience than that one I had, and this is almost entirely thanks to Rocco’s inspiration, influence, and generosity. I have far too much to thank you for, Rocco — I will always be indebted to you for all that you have taught me over the years; for your outstanding advice and your faith in me; for the many opportunities I have benefited from and the exciting ones ahead; for the literary recommendations, the Diet Dr. Peppers, and so much more. Thanks for being an amazing advisor.

Next, I would like to express my thanks to Ryan O’Donnell. I have learned a tremendous amount from Ryan, starting from my very first day at Columbia when Rocco pointed me to his survey and course notes, and later through our collaborations. Ryan is a great person on top of being an inspiring researcher and expositor, and it has been a pleasure getting to know him. I had some of my best times in graduate school over the summer I visited CMU, and I am grateful to Ryan for this and many other opportunities.

I thank Johan Håstad for inviting me spend a semester in beautiful Stockholm. Interacting with Johan was always as inspiring as it was humbling, and I benefited greatly from every one of our conversations. Thanks also to Boaz Barak for an enjoyable internship at MSR New England, and for teaching me everything I know about the Lasserre hierarchy.

I wish to also give special thanks to Andrew Wan and Dov Gordon, whose friendship helped make this such a great experience, and to Tal Malkin, for her support and candor.

I thank the members of my thesis committee, Xi Chen, Ryan O’Donnell, Rocco Servedio, Cliff Stein, and Mihalis Yannakakis, for their time and insightful comments. I have had the privilege of collaborating with many brilliant people, and I thank my co-authors for everything they have taught me. Special thanks to Eric Blais, Clément Canonne, Dominik Scheder, Justin Thaler, Andrew Wan, Karl Wimmer, David Witmer, and John Wright for all the good times we shared.
On a more personal note:

I thank my friends in New York City: Jacqui Brown, AJ Cephus, Nigel Cheong, Sonya Chandra, Kofi Edzie, Carl Jeanbart, Bryce Kirschbaum, Robyn Lym, Matteo Malinverno, Alex McCurdy, and Li Yan McCurdy. My memories of this great city are filled with the times I shared with these great people.

My teenage years were mostly spent on the diamond. I thank the team for the countless good times and their steadfast friendship, and our coach Nelson Lim for teaching me lessons that continue to serve me well today. Going back further, much of my early childhood was spent under the care of Peggy Zee, and I thank her for being the first of many great influences in my life. While I do not remember much from those years, I know she has shaped the person I am today.

And above all I thank my family for their love, without which surely none of this would have been possible.
If I had my own country
For the flag what colours would I use?
Such a myriad to choose from
I would pick red, yellow, and blue

Red, for the blood I’d spill to own it
Yellow, for the sun which shines my way
And although I am leader of this country
Blue, because I’d still have sad days

— Born Ruffians, *Red, Yellow, and Blue.*
Bibliographic Note

Much of this research has been published already, and all of it was performed jointly with other researchers. Chapter 2 of this thesis is based on the paper “New Algorithms and Lower Bounds for Testing Monotonicity” which is joint work with Xi Chen and Rocco Servedio. Chapter 3 is based on the paper “Approximate Resilience, Monotonicity, and the Complexity of Agnostic Learning” which is joint work with Dana Dachman-Soled, Vitaly Feldman, Andrew Wan, and Karl Wimmer.

Chapter 4 is based on the paper “Approximating Boolean Functions with Depth-2 Circuits”, which is joint work with Eric Blais and appeared in the Proceedings of the 28th IEEE Conference on Computational Complexity (CCC 2013). Chapter 5 is based on the paper “On DNF Approximators for Monotone Boolean Functions” which is joint work with Eric Blais, Johan Håstad, and Rocco Servedio and will appear in the Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014). Chapter 6 is based on the paper “A Composition Theorem for the Fourier Entropy–Influence Conjecture” which is joint work with Ryan O’Donnell and appeared in the Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP 2013).
Chapter 1

Introduction

1.1 Concrete complexity

Computational complexity theory studies the nature and limitations of feasible computation. It seeks to understand the role of fundamental computing resources — time, space, and randomness, among others — in the design of efficient algorithms and in computational intractability. Core open problems of the field include:

- What is the relationship between time and space? Does $P = L$ or $P = \text{PSPACE}$?
- Is randomness essential for efficient computation? Does $P = \text{BPP}$?
- Can every sequential algorithm be efficiently parallelized? Does $P = \text{NC}$?

Rigorously considering these questions within a mathematical framework necessitates a formal model of computation: what is an algorithm for solving a computational problem, and what makes the algorithm efficient?

The Turing machine is the standard model of computation in theoretical computer science. For example, we define the time complexity of a computational problem to be the number of operations necessary for a Turing machine to solve a worst-case instance, and we identify the notion of tractability with the class of problems solvable by polynomial-time Turing machines. The universality and expressiveness of the Turing machine make it a good choice — it is a generally-accepted thesis that every physically realizable computation can
be efficiently simulated on the Turing machine — but these are also the primary reasons why the core open problems of the field remain insurmountable. Strikingly, for example, we are unable to establish even super-linear time lower bounds against $\text{NP}$, a seemingly modest first step towards an eventual separation of $\text{P}$ from $\text{NP}$. This is because we remain far from understanding of what a Turing machine can and cannot compute in linear time.

Given this state of affairs it is natural to consider simpler, restricted models of computation where an algorithm is not afforded the full power of a Turing machine. This is the focus of concrete complexity, a subfield of computational complexity theory where concrete models of computation take the place of abstract Turing machines. Consider the following elementary result from concrete complexity:

*Any deterministic comparison-based algorithm for sorting a list of $N$ integers $A_1, \ldots, A_N$ must perform $\Omega(N \log N)$ operations in the worst case.*

This simple example illustrates a few hallmarks of concrete complexity:

- The algorithm is given limited access to the input and a restricted set of operations it can perform during intermediate computations. In our example above the algorithm is only permitted to make binary comparisons of the form “$A_i \geq A_j$?”.

- Computation is captured by concrete mathematical objects, and computational complexity by structural properties of these objects. In our example every comparison-based sorting algorithm has a natural representation as a binary decision tree, and its time complexity corresponds to the depth of this tree.

- Computational lower bounds, while only applying within the restricted model, are unconditional. In particular, they do not rely on unproven complexity-theoretic assumptions such as $\text{P} \neq \text{NP}$ or the Exponential Time Hypothesis.

In the next two sections we take a closer look at a few aspects of concrete complexity — the types of problems considered, the mathematical objects that arise, and the techniques involved — with Boolean functions as our guide.
CHAPTER 1. INTRODUCTION

1.2 Boolean functions

The central mathematical object in this thesis is the Boolean function,

\[ f : \{0,1\}^n \rightarrow \{0,1\}. \]

Boolean functions play a central role in several well-studied concrete models seeking to capture the complexity of basic algorithmic tasks. A few examples include:

- **Learning** [Valiant, 1984]: Given access to examples \((x^1, f(x^1)), \ldots, (x^m, f(x^m))\) labeled by \(f\), efficiently construct a good approximation \(h\) to \(f\).

- **Testing** [Rubinfeld and Sudan, 1996]: Given black-box query access to \(f\), efficiently determine whether \(f\) has a certain property \(P\).

- **Communication** [Yao, 1979]: Alice has half of the input \(x\) to \(f\), and Bob the other half. How much information do they need to exchange in order to compute \(f(x)\)?

Beyond their versatility in the modeling of basic algorithmic tasks like those above, Boolean functions are themselves a universal abstraction of computation; a Boolean function simply specifies how one output bit is determined by \(n\) input bits. Their computational complexity — the cost of computing \(f(x)\) given \(x\) — is captured by the structural complexity of mathematical objects that naturally compute them. A few examples include:

- **Circuits**: What is the minimum size and depth of any circuit computing \(f\)? How are these measures affected by the types of gates allowed in the circuit?

- **Polynomials**: What is the minimum degree and sparsity of any polynomial computing \(f\)? How are these measures affected by the underlying field (e.g. \(\mathbb{R}\) or \(\mathbb{F}_2\))?  

- **Halfspaces**: Can \(f\) be computed by a halfspace? That is, can we write \(f(x) = \text{sign}(w \cdot x - \theta)\) for some \(w \in \mathbb{R}^n\) and \(\theta \in \mathbb{R}\)? More generally, what is the smallest integer \(d\) such that \(f(x) = \text{sign}(p(x))\) for a degree-\(d\) polynomial \(p\)?

In this thesis we study Boolean functions from both vantage points, considering both the computational complexity of algorithmic tasks involving Boolean functions and the
structural complexity of Boolean functions themselves. In Parts I and II we present new algorithms and lower bounds for testing and learning Boolean functions, and in Part III we prove new structural theorems about circuits and polynomials computing Boolean functions. As we will see these two directions are closely intertwined; in particular, the computational results in the first two parts build heavily on an improved understanding of the mathematical structure of Boolean functions.

1.3 Analytic methods

The results in this thesis are obtained by applying a range of tools from different branches of mathematics, with analytic methods — Fourier analysis, probability theory, and approximation theory — playing a central role.

The analysis of Boolean functions is concerned with properties of Boolean functions viewed as real polynomials via their Fourier transform. Introduced into theoretical computer science by the work of Kahn, Kalai, and Linial [Kahn et al., 1988], this simple point of view is by now a mainstay of concrete complexity and has transformed areas such as property testing, learning theory, and the hardness of approximation. Its broad influence is mainly due to the many parallels that have been established between combinatorial properties of a Boolean function — linearity, monotonicity, noise sensitivity, etc. — and analytic properties of its Fourier spectrum. These connections allow us to draw on a range of analytic tools to tackle computational problems that tend to be intrinsically combinatorial in nature. A few examples include:

- A Boolean function computes a linear \( \mathbb{F}_2 \)-polynomial if and only if its Fourier expansion consists of a single monomial. A generalization of this simple fact underlies the analysis of a constant-query tester for the linearity of Boolean functions [Blum et al., 1993; Bellare et al., 1996].

- A monotone Boolean function has almost all of its Fourier mass concentrated on coefficients of low degree. This is the basis for our best-known algorithms for PAC learning monotone Boolean functions under the uniform distribution [Bshouty and Tamon, 1996; Servedio, 2004b; O’Donnell and Servedio, 2008].
• The circuit complexity of an $n$-variable Boolean function $f$ composed with $n$ copies of a Boolean function $g$ is determined by the circuit complexity of $g$ and the noise sensitivity of $f$. This characterization, proved using Fourier-analytic methods, initiated our study of hardness amplification within $\text{NP}$ [O’Donnell, 2004].

In this thesis we further develop the analytic toolkit for studying Boolean functions, and we apply these tools to establish new and stronger connections between their combinatorial structure and Fourier spectrum. We leverage these structural theorems to obtain new computational results, both algorithmic upper bounds and complexity-theoretic lower bounds, in property testing, learning theory, and circuit complexity.

1.4 Outline of this thesis

Part I: Property Testing

In Chapter 2 we consider the problem of Property Testing — efficiently determining whether a large, often high-dimensional, mathematical object has a certain property — within the elegant framework of Rubinfeld and Sudan [Rubinfeld and Sudan, 1996]. We study the classical problem of testing whether an unknown Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is monotone versus $\varepsilon$-far from every monotone function, and our two main results are a new lower bound and a new algorithm for this well-studied problem. First, we prove an $\tilde{\Omega}(n^{1/5})$ lower bound on the query complexity of any non-adaptive two-sided error algorithm for testing whether an unknown Boolean function $f$ is monotone versus constant-far from monotone. This gives an exponential improvement over the previous lower bound of $\Omega(\log n)$ due to Fischer et al. [Fischer et al., 2002]. We also show that the same lower bound holds for monotonicity testing of Boolean-valued functions over hypergrid domains $\{1, \ldots, m\}^n$ for all $m \geq 2$. Second, we give an $\tilde{O}(n^{5/6})\text{poly}(1/\varepsilon)$-query algorithm that tests whether an unknown Boolean function $f$ is monotone versus $\varepsilon$-far from monotone. Our algorithm, which is non-adaptive and makes one-sided error, is a modified version of the algorithm of Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013a], which makes $\tilde{O}(n^{7/8})\text{poly}(1/\varepsilon)$ queries.
**Part II: Computational Learning Theory**

In **Chapter 3** we consider the problem of *Agnostic Learning* — efficient learning in the presence of harsh adversarial noise — within Kearns’ well-studied statistical query variant [Kearns et al., 1994] of Valiant’s Probably Approximately Correct model [Valiant, 1984]. Our main result in this chapter is a characterization of the statistical query (SQ) complexity of agnostic learning via the analytic notion of *approximate resilience* of Boolean functions. Roughly speaking, we show that if all functions in a class $\mathcal{C}$ are far from being $d$-resilient then $\mathcal{C}$ can be learned agnostically in time $n^{O(d)}$, and conversely, if $\mathcal{C}$ contains a function close to being $d$-resilient then the SQ complexity of agnostically learning $\mathcal{C}$ is at least $n^{\Omega(d)}$. Our characterization implies that $\ell_1$-approximability by low-degree polynomials, known to be sufficient for agnostic learning over product distributions [Kalai et al., 2008], is in fact necessary. As an application we give an optimal lower bound showing that no SQ algorithm can efficiently agnostically learn monotone $k$-juntas for any $k = \omega(1)$ and any constant error less than $1/2$.

**Part III: Boolean Function Complexity**

In the third part of this thesis we study the structural complexity of mathematical objects — Boolean circuits and real polynomials — that compute Boolean functions.

In **Chapters 4 and 5** we initiate a systematic study of the tradeoffs between accuracy and efficiency in Boolean circuit complexity, focusing on disjunctive normal form (DNF) formulas, which are among the most basic types of circuits. We begin in **Chapter 4** by exploring two main directions: universal bounds on the approximability of all Boolean functions, and the approximability of the parity function. In the first direction, our main positive results are the first non-trivial universal upper bounds on approximability by DNFs, showing that:

- Every Boolean function can be $\varepsilon$-approximated by a DNF of size $O_\varepsilon(2^n / \log n)$.
- Every Boolean function can be $\varepsilon$-approximated by a DNF of width $c_\varepsilon n$, where $c_\varepsilon < 1$.

In the second direction our main positive result is the construction of an explicit DNF that approximates the parity function, showing that:
• \( \text{PAR}_n \) can be \( \varepsilon \)-approximated by a DNF of size \( 2^{(1-2\varepsilon)n} \) and width \( (1-2\varepsilon)n \).

We continue this study in Chapter 5, turning our attention to the complexity of approximating monotone Boolean functions with DNF formulas. Our first result in this chapter is an explicit construction of DNF approximators for arbitrary monotone functions achieving one-sided error: we show that every monotone \( f \) can be \( \varepsilon \)-approximated by a DNF \( g \) of size \( 2^{n-\Omega(\sqrt{n})} \) satisfying \( g(x) \leq f(x) \) for all \( x \in \{0,1\}^n \). Next we study the power of negations in DNF approximators for monotone functions. We exhibit monotone functions for which non-monotone DNF formulas perform better than monotone ones, giving separations with respect to both DNF size and width.

In Chapter 6 we study the Fourier Entropy-Influence (FEI) conjecture, a long-standing open problem in the analysis of Boolean functions with significant applications in learning theory, the theory of pseudorandomness, and random graph theory. The conjecture seeks to relate two fundamental measures of Boolean function complexity: spectral entropy and total influence. Our main result in this chapter is a composition theorem for the FEI conjecture. We show that if \( g_1, \ldots, g_k \) are functions over disjoint sets of variables satisfying the conjecture, and if the Fourier transform of \( F \) taken with respect to the product distribution with biases \( E[g_1], \ldots, E[g_k] \) satisfies the conjecture, then their composition \( F(g_1(x^1), \ldots, g_k(x^k)) \) satisfies the conjecture. As an application we show that the FEI conjecture holds for read-once formulas over arbitrary gates of bounded arity, extending a recent result [O’Donnell et al., 2011] which proved it for read-once decision trees. Our techniques also yield an explicit function with the largest known ratio of \( C \geq 6.278 \) between total influence and spectral entropy, improving on the previous lower bound of 4.615.

Finally in Chapter 7 we close with a brief discussion of future research directions.