

Research Statement

Parikshit Gopalan

My research focuses on fundamental algebraic problems such as polynomial reconstruction and interpolation arising from various areas of theoretical computer science. My main algorithmic contributions include the first algorithm for list-decoding a well-known family of codes called Reed-Muller codes [13], and the first algorithms for agnostically learning parity functions [3] and decision trees [11] under the uniform distribution. On the complexity-theoretic side, my contributions include the best-known hardness results for reconstructing low-degree multivariate polynomials from noisy data [12] and the discovery of a connection between representations of Boolean functions by polynomials and communication complexity [2].

1 Introduction

Many important recent developments in theoretical computer science, such as probabilistic proof checking, deterministic primality testing and advancements in algorithmic coding theory, share a common feature: the extensive use of techniques from algebra. My research has centered around the application of these methods to problems in Coding theory, Computational learning, Hardness of approximation and Boolean function complexity.

While at first glance, these might seem like four research areas that are not immediately related, there are several beautiful connections between these areas. Perhaps the best illustration of these links is the noisy parity problem where the goal is to recover a parity function from a corrupted set of evaluations. The seminal Goldreich-Levin algorithm solves a version of this problem; this result initiated the study of list-decoding algorithms for error-correcting codes [5]. An alternate solution is the Kushilevitz-Mansour algorithm [19], which is a crucial component in algorithms for learning decision trees and DNFs [17]. Håstad's ground-breaking work on the hardness of this problem has revolutionized our understanding of inapproximability [16]. All these results rely on insights into the Fourier structure of Boolean functions.

As I illustrate below, my research has contributed to a better understanding of these connections, and yielded progress on some important open problems in these areas.

2 Coding Theory

The broad goal of coding theory is to enable meaningful communication in the presence of noise, by suitably encoding the messages. The natural algorithmic problem associated with this task is that of decoding or recovering the transmitted message from a corrupted encoding. The last twenty years have witnessed a revolution with the discovery of several powerful decoding algorithms for well-known families of error-correcting codes. A key role has been played by the notion of list-decoding; a relaxation of the classical decoding problem where we are willing to settle for a small list of candidate transmitted messages rather than insisting on a unique answer. This relaxation allows one to break the classical *half the minimum distance* barrier for decoding error-correcting codes. We now know powerful list-decoding algorithms for several important code families, these algorithms have also made a huge impact on complexity theory [5, 15, 23].

List-Decoding Reed-Muller Codes: In recent work with Klivans and Zuckerman, we give the first such list-decoding algorithm for a well-studied family of codes known as Reed-Muller codes, obtained from low-degree polynomials over the finite field \mathbb{F}_2 [13]. The highlight of this work is that our algorithm is able to tolerate error-rates which are much higher than what is known as the Johnson bound in coding theory. Our results imply new combinatorial bounds on the error-correcting capability of these codes. While Reed-Muller codes have been studied extensively in both coding theory and computer science communities, our result is the first to show that they are resilient to remarkably high error-rates. Our algorithm is based on a novel view of the Goldreich-Levin algorithm as a reduction from list-decoding to unique-decoding; our view readily extends to polynomials of arbitrary degree over any field. Our result complements recent work on the Gowers norm, showing that Reed-Muller codes are testable up to large distances [21].

Hardness of Polynomial Reconstruction: In the polynomial reconstruction problem, one is asked to recover a low-degree polynomial from its evaluations at a set of points and some of the values could be incorrect. The reconstruction problem is ubiquitous in both coding theory and computational learning. Both the Noisy parity problem and the Reed-Muller decoding problem are instances of this problem. In joint work with Khot and Saket, we address the complexity of this problem and establish the first hardness results for multivariate polynomials of arbitrary degree [12]. Previously, the only hardness known was for degree 1, which follows from the celebrated work of Håstad [16]. Our work introduces a powerful new algebraic technique called global folding which allows one to bypass a module called consistency testing that is crucial to most hardness results. I believe this technique will find other applications.

Average-Case Hardness of NP: Algorithmic advances in decoding of error-correcting codes have helped us gain a deeper understand of the connections between worst-case and average case complexity [23, 24]. In recent work with Guruswami, we use this paradigm to explore the average-case complexity of problems in NP against algorithms in P [8]. We present the first hardness amplification result in this setting by giving a construction of an error-correcting code where most of the symbols can be recovered correctly from a corrupted codeword by a deterministic algorithm that probes very few locations in the codeword. The novelty of our work is that our decoder is deterministic, whereas previous algorithms for this task were all randomized.

3 Computational Learning

Computational learning aims to understand the algorithmic issues underlying how we learn from examples, and to explore how the complexity of learning is influenced by factors such as the ability to ask queries and the possibility of incorrect answers. Learning algorithms for a class of concept typically rely on understanding the structure of that concept class, which naturally ties learning to Boolean function complexity. Learning in the presence of noise has several connections to decoding from errors. My work in this area addresses the learnability of basic concept classes such as decision trees, parities and halfspaces.

Learning Decision Trees Agnostically: The problem of learning decision trees is one of the central open problems in computational learning. Decision trees are also a popular hypothesis class in practice. In recent work with Kalai and Klivans, we give a query algorithm for learning decision trees with respect to the uniform distribution on inputs in the agnostic model: given black-box access to an *arbitrary* Boolean function, our algorithm finds a hypothesis that agrees with it on almost as many inputs as the best decision tree [11]. Equivalently, we can learn decision trees even when the data is corrupted adversarially; this is the first polynomial-time algorithm for learning decision trees in a harsh noise model. Previous decision-tree learning algorithms applied only to the noiseless setting. Our algorithm can be viewed as the agnostic analog of the

Kushilevitz-Mansour algorithm [19]. The core of our algorithm is a procedure to implicitly solve a convex optimization problem in high dimensions using approximate gradient projection.

The Noisy Parity Problem: The Noisy parity problem has come to be widely regarded as a hard problem. In work with Feldman *et al.*, we present evidence supporting this belief [3]. We show that in the setting of learning from random examples (without queries), several outstanding open problems such as learning juntas, decision trees and DNFs reduce to restricted versions of the problem of learning parities with random noise. Our result shows that in some sense, noisy parity captures the gap between learning from random examples and learning with queries, as it is believed to be hard in the former setting and is known to be easy in the latter. On the positive side, we present the first non-trivial algorithm for the noisy parity problem under the uniform distribution in the adversarial noise model. Our result shows that somewhat surprisingly, adversarial noise is no harder to handle than random noise.

Hardness of Learning Halfspaces: The problem of learning halfspaces is a fundamental problem in computational learning. One could hope to design algorithms that are robust even in the presence of a few incorrectly labeled points. Indeed, such algorithms are known in the setting where the noise is random. In work with Feldman *et al.*, we show that the setting of adversarial errors might be intractable: given a set of points where 99% are correctly labeled by some halfspace, it is NP-hard to find a halfspace that correctly labels even 51% of the points [3].

4 Prime versus Composite problems

My thesis work focuses on new aspects of an old and famous problem: the difference between primes and composites. Beyond basic problems like primality and factoring, there are many other computational issues that are not yet well understood. For instance, in circuit complexity, we have excellent lower bounds for small-depth circuits with mod 2 gates, but the same problem for circuits with mod 6 gates is wide open. Likewise in combinatorics, set systems where sizes of the sets need to satisfy certain modular conditions are well studied. Again the prime case is well understood, but little is known for composites. In all these problems, the algebraic techniques that work well in the prime case break down for composites.

Boolean function complexity: Perhaps the simplest class of circuits for which we have been unable to show lower bounds is small-depth circuits with And, Or and Mod m gates where m is composite; indeed this is one of the frontier open problems in circuit complexity. When m is prime, such bounds were proved by Razborov and Smolensky [20, 22]. One reason for this gap is that we do not fully understand the computational power of polynomials over composites; Barrington *et al* were the first to show that such polynomials are surprisingly powerful [1]. In joint work with Bhatnagar and Lipton, we solve an important special case: when the polynomials are symmetric in their variables [2]. We show an equivalence between computing Boolean functions by symmetric polynomials over composites and multi-player communication protocols, which enables us to apply techniques from communication complexity and number theory to this problem. We use these techniques to show tight degree bounds for various classes of functions where no bounds were known previously. Our viewpoint simplifies previously known results in this area, and reveals new connections to well-studied questions about Diophantine equations.

Explicit Ramsey Graphs: A basic open problem regarding polynomials over composites is: *Can asymmetry in the variables help us compute a symmetric function with low degree?* I show a connection between this question and an important open problem in combinatorics, which is to explicitly construct Ramsey graphs or graphs with no large cliques and independent sets [6]. While good Ramsey graphs are known to exist by probabilistic arguments, explicit constructions have proved elusive. I propose a new algebraic framework for constructing Ramsey graphs and showed how

several known constructions can all be derived from this framework in a unified manner. I show that all known constructions rely on symmetric polynomials, and that such constructions cannot yield better Ramsey graphs. Thus the question of symmetry versus asymmetry of variables is precisely the barrier to better constructions by such techniques.

Interpolation over Composites: A basic problem in computational algebra is polynomial interpolation, which is to recover a polynomial from its evaluations. Interpolation and related algorithmic tasks which are easy for primes become much harder, even intractable over composites. This difference stems from the fact that over primes, the number of roots of a polynomial is bounded by the degree, but no such theorem holds for composites. In lieu of this theorem I presented an algorithmic bound; I show how to compute a bound on the degree of a polynomial given its zero set [7]. I use this to give the first optimal algorithms for interpolation, learning and zero-testing over composites. These algorithms are based on new structural results about the zeroes of polynomials. These results were subsequently useful in ruling out certain approaches for better Ramsey constructions [6].

5 Other Research Highlights

My other research work spans areas of theoretical computer science ranging from algorithms for massive data sets to computational complexity. I highlight some of this work below.

Data Stream Algorithms: Algorithmic problems arising from complex networks like the Internet typically involve huge volumes of data. This has led to increased interest in highly efficient algorithmic models like sketching and streaming, which can meaningfully deal with such massive data sets. A large body of work on streaming algorithms focuses on estimating how sorted the input is. This is motivated by the realization that sorting the input is intractable in the one-pass data stream model. In joint work with Krauthgamer, Jayram and Kumar, we presented the first sub-linear space data stream algorithms to estimate two well-studied measures of sortedness: the distance from monotonicity (or Ulam distance for permutations), and the length of the Longest Increasing Subsequence or LIS.

In more recent work with Anna Gál, we prove optimal lower bounds for estimating the length of the LIS in the data-stream model [4]. This is established by proving a direct-sum theorem for the communication complexity of a related problem. The novelty of our techniques is the model of communication that they address. As a corollary, we obtain a separation between two models of communication that are commonly studied in relation to data stream algorithms.

Structural Properties of SAT solutions: The solution space of random SAT formulae has been studied with a view to better understanding connections between computational hardness and phase transitions from satisfiable to unsatisfiable. Recent algorithmic approaches rely on connectivity properties of the space and break down in the absence of connectivity. In joint work with Kolaitis, Maneva and Papadimitriou, we consider the problem: *Given a Boolean formula, do its solutions form a connected subset of the hypercube?* We classify the worst-case complexity of various connectivity properties of the solution space of SAT formulae in Schaefer's framework [14]. We show that the jump in the computational hardness is accompanied by a jump in the diameter of the solution space from linear to exponential.

Complexity of Modular Counting Problems: In joint work with Guruswami and Lipton, we address the complexity of counting the roots of a multivariate polynomial over a finite field \mathbb{F}_q modulo some number r [9]. We establish a dichotomy showing that the problem is easy when r is a power of the characteristic of the field and intractable otherwise. Our results give several examples of problems whose decision versions are easy, but the modular counting version is hard.

6 Future Research Directions

My broad research goal is to gain a complete understanding of the complexity of problems arising in coding theory, computational learning and related areas; I believe that the right tools for this will come from Boolean function complexity and hardness of approximation. Below I outline some of the research directions I would like to pursue in the future.

List-decoding algorithms have allowed us to break the unique-decoding barrier for error-correcting codes. It is natural to ask if one can perhaps go beyond the list-decoding radius and solve the problem of finding the codeword nearest to a received word at even higher error rates. On the negative side, we do not currently know any examples of codes where one can do this. But I think that recent results on Reed-Muller codes do offer some hope [13, 21]. Algorithms for solving the nearest codeword problem if they exist, could also have exciting implications in computational learning. There are concept classes which are well-approximated by low-degree polynomials over finite fields lying just beyond the threshold of what is currently known to be learnable efficiently [20, 22]. Decoding algorithms for Reed-Muller codes that can tolerate very high error rates might present an approach to learning such concept classes.

One of the challenges in algorithmic coding theory is to determine whether known algorithms for list-decoding Reed-Solomon codes [15] and Reed-Muller codes [13, 23] are optimal. This raises both computational and combinatorial questions. I believe that my work with Khot *et al.* represents a good first step towards understanding the complexity of the decoding/reconstruction problem for multivariate polynomials. Proving similar results for univariate polynomials is an excellent challenge which seems to require new ideas in hardness of approximation.

There is a large body of work proving strong NP-hardness results for problems in computational learning. However, all such results only address the proper learning scenario where the learning algorithm is restricted to produce a hypothesis from some particular class \mathcal{H} which is typically the same as the concept class \mathcal{C} . In contrast, known learning algorithms are mostly improper algorithms which could use more complicated hypotheses. For hardness results that are independent of the hypothesis \mathcal{H} used by the algorithm, one currently has to resort to cryptographic assumptions. In ongoing work with Guruswami and Raghavendra, we are investigating the possibility of proving NP-hardness for improper learning.

Finally, I believe that there are several interesting directions to explore in the agnostic learning model. An exciting insight in this area comes from the work of Kalai *et al.* who show that ℓ_1 regression is a powerful tool for noise-tolerant learning [18]. A powerful paradigm in computational learning is to prove that the concept has some kind of polynomial approximation and then recover the approximation. Algorithms based on ℓ_1 regression require a weaker polynomial approximation in comparison with previous algorithms (which use ℓ_2 regression), but use more powerful machinery for the recovery step. Similar ideas might allow us to extend the boundaries of efficient learning even in the noiseless model; this is a possibility I am currently exploring.

Having worked in areas ranging from data stream algorithms to Boolean function complexity, I view myself as both an algorithm designer and a complexity theorist. I have often found that working on one aspect of a problem gives insights into the other; indeed much of my work has originated from such insights ([12] and [13], [10] and [4], [6] and [7]). I find that this is increasingly the case across several areas in theoretical computer science. My aim is to maintain this balance between upper and lower bounds in my future work.

References

- [1] D. A. Barrington, R. Beigel, and S. Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994.

- [2] N. Bhatnagar, P. Gopalan, and R. J. Lipton. Symmetric polynomials over \mathbb{Z}_m and simultaneous communication protocols. *Journal of Computer & System Sciences (special issue for FOCS'03)*, 72(2):450–459, 2003.
- [3] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc. 47th IEEE Symp. on Foundations of Computer Science (FOCS'06)*, 2006.
- [4] A. Gál and P. Gopalan. Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. In *Proc. 48th IEEE Symp. on Foundations of Computer Science (FOCS'07)*, 2007.
- [5] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proc. 21st ACM Symposium on the Theory of Computing (STOC'89)*, pages 25–32, 1989.
- [6] P. Gopalan. Constructing Ramsey graphs from Boolean function representations. In *Proc. 21st IEEE symposium on Computational Complexity (CCC'06)*, 2006.
- [7] P. Gopalan. Query-efficient algorithms for polynomial interpolation over composites. In *Proc. 17th ACM-SIAM symposium on Discrete algorithms (SODA'06)*, 2006.
- [8] P. Gopalan and V. Guruswami. Deterministic hardness amplification via local GMD decoding. *Submitted to 23rd IEEE Symp. on Computational Complexity (CCC'08)*, 2008.
- [9] P. Gopalan, V. Guruswami, and R. J. Lipton. Algorithms for modular counting of roots of multivariate polynomials. In *Proc. Latin American Symposium on Theoretical Informatics (LATIN'06)*, 2006.
- [10] P. Gopalan, T.S. Jayram, R. Krauthgamer, and R. Kumar. Estimating the sortedness of a data stream. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 2007.
- [11] P. Gopalan, A. T. Kalai, and A. R. Klivans. Agnostically learning decision trees. In *Proc. 40th ACM Symp. on Theory of Computing (STOC'08)*, 2008.
- [12] P. Gopalan, S. Khot, and R. Saket. Hardness of reconstructing multivariate polynomials over finite fields. In *Proc. 48th IEEE Symp. on Foundations of Computer Science (FOCS'07)*, 2007.
- [13] P. Gopalan, A. R. Klivans, and D. Zuckerman. List-decoding Reed-Muller codes over small fields. In *Proc. 40th ACM Symp. on Theory of Computing (STOC'08)*, 2008.
- [14] P. Gopalan, P. G. Kolaitis, E. N. Maneva, and C. H. Papadimitriou. Computing the connectivity properties of the satisfiability solution space. In *Proc. 33rd Intl. Colloquium on Automata, Languages and Programming (ICALP'06)*, 2006.
- [15] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and Algebraic-Geometric codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.
- [16] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
- [17] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.
- [18] A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio. Agnostically learning halfspaces. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, pages 11–20, 2005.
- [19] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal of Computing*, 22(6):1331–1348, 1993.

- [20] A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. *Mathematical Notes of the Academy of Science of the USSR*, (41):333–338, 1987.
- [21] A. Samorodnitsky. Low-degree tests at large distances. In *Proc. 39th ACM Symposium on the Theory of Computing (STOC'07)*, pages 506–515, 2007.
- [22] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. *Proc. 19th Annual ACM Symposium on Theoretical Computer Science, (STOC'87)*, pages 77–82, 1987.
- [23] M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.
- [24] L. Trevisan. List-decoding using the XOR lemma. In *Proc. 44th IEEE Symposium on Foundations of Computer Science (FOCS'03)*, pages 126–135, 2003.