Mathematisches Forschungsinstitut Oberwolfach

Report No. 54/2012

DOI: 10.4171/OWR/2012/54

Complexity Theory

Organised by
Peter Bürgisser, Paderborn
Oded Goldreich, Rehovot
Madhu Sudan, Cambridge MA
Salil Vadhan, Cambridge MA

11th November – 17th November 2012

Abstract. Computational Complexity Theory is the mathematical study of the intrinsic power and limitations of computational resources like time, space, or randomness. The current workshop focused on recent developments in various sub-areas including arithmetic complexity, Boolean complexity, communication complexity, cryptography, probabilistic proof systems, and pseudorandomness. Many of the developments are related to diverse mathematical fields such as algebraic geometry, combinatorial number theory, probability theory, representation theory, and the theory of error-correcting codes.

Mathematics Subject Classification (2000): 68-06, 68Q01, 68Q10, 68Q15, 68Q17, 68Q25, 94B05, 94B35.

Introduction by the Organisers

The workshop *Complexity Theory* was organized by Peter Bürgisser (Universität Paderborn), Oded Goldreich (Weizmann Institute), Madhu Sudan (MIT and Microsoft Research), and Salil Vadhan (Harvard). The workshop was held on November 11th–17th 2012, and attended by approximately 50 participants spanning a wide range of interests within the field of Computational Complexity. The plenary program, attended by all participants, featured fifteen long lectures, an open problem session, and ten short (5-minute) reports mostly by students and postdocs. In addition, intensive interaction took place in smaller groups.

The Oberwolfach Meeting on Complexity Theory is marked by a long tradition and a continuous transformation. Originally starting with a focus on algebraic and Boolean complexity, the meeting has continuously evolved to cover a wide variety of areas, most of which were not even in existence at the time of the first meeting (in 1972). While inviting many of the most prominent researchers in the field, the organizers try to identify and invite a fair number of promising young researchers.

Computational complexity (a.k.a. complexity theory) is a central field of computer science with a remarkable list of celebrated achievements as well as a vibrant research activity. The field is concerned with the study of the *intrinsic complexity* of computational tasks, and this study tends to *aim at generality*: it focuses on natural computational resources, and considers the effect of limiting these resources on the class of problems that can be solved. Computational complexity is related to and has substantial interaction with other areas of mathematics such as algebra, analysis, combinatorics, geometry, number theory, optimization, probability theory, and quantum computation.

The workshop focused on several sub-areas of complexity theory and its nature may be best illustrated by a brief survey of some of the meeting's highlights.

Matrix Multiplication. The Oberwolfach meeting on complexity theory that took place in 1979 is famous for Strassen's presentation of his landmark subcubic time algorithm for matrix multiplication and a sequence of improvements that followed via interaction of several participants in that meeting. In connection with that tradition, Virginia Vassilevska Williams presented her recent algorithmic improvement that breaks the record set in 1987 by Coppersmith and Winograd. Her improvement is based on structural results regarding the mathematical objects that arise in the Coppersmith and Winograd algorithm, which lend themselves to an automatic search for better objects (using convex and nonconvex optimization).

Chris Umans reported on the state of his on-going project for designing better matrix multiplication algorithms. In particular, a generalization of the group-theoretic approach to using "coherent configurations" seems to facilitate further progress in this project, which has a potential of obtaining almost-optimal (i.e., quadratic-time) algorithms. So far, however, the group-theoretic approach was only able to get close to the best upper bounds known (but did not quite meet them, let alone supersede them).

Boolean Circuit Lower Bounds. The project of establishing circuit lower bounds calls for presenting (relatively) explicit functions that cannot be computed within limited computational resources. Ryan Williams presented the most significant progress in circuit lower bounds since the 1980s, proving that there exists a function in non-deterministic exponential-time (i.e., in the complexity class NE) that cannot be computed by polynomial-size Boolean circuits of constant depth with modular gates (i.e., the class ACC). The breakthrough is in proving a lower bound for less limited circuits than those considered in the past, although the level of explicitness (i.e., being in NE) is relatively weaker than in previous results.

Interestingly, his lower bound is obtained by designing an algorithm, one that slightly improves over the obvious algorithm for testing satisfiability of ACC circuits. Combined with the contradiction hypothesis (by which NE is in ACC), this algorithm yields an impossible speed-up for the class NE (i.e., one that contradicts a known hierarchy theorem). This suggests that the fact that certain pseudorandom generators (PRGs) imply circuit lower bounds that are better than currently known does not necessarily mean that obtaining such PRGs is hopeless; it may just be a way to establishing new lower bounds.

Pseudorandom Generators. Pseudorandom generators (PRGs) are deterministic algorithms that stretch short random seeds into longer (pseudorandom) sequences that look random to distinguishers that are confined to certain complexity classes. Various notions of PRGs differ by the class of distinguishers that they fool as well as the efficiency of the PRG itself and the amount of stretch, and in some cases the construction of PRGs is related to the existence of related lower bounds (or to the assumption that such lower bounds hold).

Raghu Meka surveyed recent progress made in the study of unconditional pseudorandom generators; that is, PRGs that can be constructed without relying on any unproved computational hardness conjectures. One research direction led to optimal conversion of known results regarding computational hardness into pseudorandomness with respect to the corresponding classes (i.e., the classes capturing corresponding computational resources). The results obtained in this direction are based on a novel integration of a lower bound technique (i.e., the technique of random restrictions) in the context of PRGs. A second direction led to an almost polynomial-time deterministic algorithm for approximating the number of solutions to a given DNF formula.

Turning to hardness and pseudorandomness with respect to any efficient (i.e., probabilistic polynomial-time) computation, Benny Applebaum surveyed the construction of PRGs that are extremely easy to compute in the sense that each output bit depends on a constant number of input bits (i.e., the class NC0). One recent work, which he mentioned, shows that the assumption that certain NC0 functions are easy to compute but hard to invert (i.e., that these functions are one-way functions) implies that related functions are PRGs.

Homomorphic Encryption. A fully homomorphic encryption scheme is one that allows for arbitrary manipulation of ciphertexts without decrypting them; that is, for any polynomial-time computable function f, one can efficiently obtain an encryption of $f(x)$ when given an encryption of x (without being able to decrypt). The notion, suggested in the 1980s, was considered unimplementable till a few years ago, when first evidence to its feasibility was given. Zvika Brakerski presented the most up-to-date evidence for this feasibility, relying on the conjectured hardness of learning parity with noise.

Delegating Computation and Complexity theory. The possibility of delegating computation to untrusted parties relies on the possibility of verifying the correctness of such computation. For this to make sense, verification ought to be significantly faster than the original computation, whereas convincing the verifier (i.e., the proving task) should remain feasible (or relatively feasible in comparison to the original computation). Guy Rothblum presented recent progress in this direction, presenting both an interactive proof system and an interactive proof of proximity (in the spirit of property testing) for problems in the complexity class NC.

Differential Privacy and Complexity theory. This area is concerned with the study of trade-offs between the utility available from a "sanitized" data base and the level of privacy (of individual records) preserved by such a mechanism. In principle, one should be able to extract global statistics while violating privacy to a very small extent. Results of this type started to appear less than a decade ago and a more systematic study arose in the last few years.

Moritz Hardt provided a survey of recent progress in this area, emphasizing complexity-theoretic aspects such as highly non-trivial composition theorems, connections to computational learning theory, and open questions regarding the computational complexity of some problems in differential privacy.

Machine learning (albeit in the unsupervised rather than supervised learning context) was the focus of Sanjeev Arora's presentation, which highlighted algorithmic progress and challenges in this area.

The Unique Games Conjecture. Introduced a decade ago, the unique games conjecture (UGC) states that constraint satisfaction problems involving two variables and constraints that correspond to a matching between values are hard to approximate in an extreme sense (i.e., it is infeasible to distinguish instances in which almost all constraints can be simultaneously satisfied from instances in which only few constraints can be simultaneously satisfied). Boaz Barak presented a survey of recent research on the UGC, presenting both positive and negative circumstantial evidence for the validity of UGC.

Communication Complexity. Anup Rao surveyed recent progress in communication complexity that is based on the notion of Interactive Information Complexity (IIC). The key point is that IIC allows to prove that resources must be increased when trying to solve several independent instances. This is done by showing that a protocol that solves the multi-instance problem can be transformed into a much more efficient protocol that solves a single instance, where the "complexity shrinkage" is obtained by noting that the communication in the multi-instance protocol carries relatively little information on a typical instance.

Communication complexity was also pivotal in David Steurer's presentation, where it was used to prove exponential lower bounds on the size of linear programs that solve certain natural optimization problems.

Additive Combinatorics and its Applications to Complexity. Noga Ron-Zewi's presentation focused on the Polynomial Freiman-Ruzsa conjecture and its applications to complexity theory, which are derived via the notion of approximate duality. The applications she highlighted are to the construction of two-source randomness extractors and towards proving the Log-Rank Conjecture in communication complexity.

Computational Aspects of Coding Theory. The method of multiplicities is based on the observation that the number of roots of multivariate polynomial, counted with multiplicities, does not exceed the bound commonly used for counting roots without multiplicities. This bound is meaningful even when the total degree exceeds the size of the base field. This observation can be used in the analysis of various constructs that are based on multivariate polynomials as well as in the actual construction of locally decodable codes. Shubhangi Saraf's presentation focused on the latter case, aka Multiplicity Codes, where a multivariate polynomial (over a finite field) is encoded by its evaluation as well as by it derivatives at all points of the domain. It is remarkable that multiplicity codes, while easy to construct, can be efficiently be decoded up to the list decoding capacity.

Lower Bounds for Arithmetic Circuits Valiant conjectured in 1979 that the permanent per_n of an n by n matrix cannot be computed by arithmetic circuits of size polynomial in n. This fundamental problem of algebraic complexity is often considered the arithmetic version of P versus NP. In his talk, Pascal Koiran explained the recent progress around this question. A relatively new insight is that attention may be restricted to circuits of depth four, which means considering sums of products of sparse polynomials: more specifically, Valiant's Conjecture would follow from a lower bound $2^{\omega(\sqrt{n}\log^2 n)}$ for the size of arithmetic circuits of depth four that compute per_n. Very recently, the lower bound $2^{\Omega(\sqrt{n})}$ was obtained, which seems close to the objective. The proof of this exciting result was presented in detail in a special session by Neeraj Kayal. It relies on considering the growth of the Hilbert function of permanental ideals. There seems potential for further improvements.

Another remarkable recent result is a connection of Valiant's Conjecture to a question concerning the number of real zeros of polynomials. More specifically, the *Real Tau Conjecture* claims that the number of real zeros of a polynomial given by a depth four circuit is bounded by a polynomial in the size of the circuit. Koiran proved that this conjecture implies Valiant's Conjecture. The Real Tau Conjecture is currently wide open: There is evidence that it holds for random polynomials. Unlike its cousin, Shub and Smale's Tau Conjecture, it is not of a number-theoretic nature and so there is hope that it can be successfully attacked using tools from analysis. Some progress has been made in this direction using Wronskian determinants.

An open problem session. As part of the plenary session, Boaz Barak has organized an open problem session, which included the following presentations:

- Results and conjectures which explain why different optimization and constraint-satisfaction problems (such as 2SAT vs. 3SAT) have different complexities (Prasad Raghavendra).
- Frontiers in the construction of expander graphs (Omer Reingold).
- A conjecture regarding the inapproximability of constraint satisfaction problems where the constraints are weighted majority functions with very unbalanced weights (Johan Hastad).
- The next step in the construction of pseudorandom generators that fool depth-two circuits, or the seemingly last step that yields no new lower bound (Luca Trevisan).

In all cases, the problem was presented in its wider context, while stressing the conceptual importance of the question to this wider context.

Informal specialized sessions. Besides the formal plenary program, intense interaction between the participants took place in smaller groups. Part of these took place in the form of specialized sessions, featuring the following presentations.

- Hardness results in differential privacy (by Salil Vadhan, in continuation of Moritz Hardt's plenary presentation).
- Direct products in communication complexity (by Anup Rao, in continuation of his plenary presentation).
- Open discussion on PCPs.
- A session on Extractors, Expanders and PRGs, with talks on
  - algebraic expanders (survey by Amir Yehudayoff),
  - algebraic SL vs L (by Michael Forbes),
  - characterizing pseudoentropy (by Salil Vadhan),
  - new extractors using multiplicities (by Chris Umans, in continuation of Shubhangi Saraf's plenary presentation).
- A session on Cryptography, featuring talks on
  - functional encryption and reusable garbled circuits (by Shafi Goldwasser),
  - modulus-dimension trade-offs in the Learning with Errors problem (by Zvika Brakerski)
- Parallel Repetition Theorem for projection games (by Irit Dinur).
- Population Recovery (by Avi Wigderson).
- Explicit lower bounds via geometric complexity theory (by Christian Ikenmeyer).
- Lower bounds for depth four arithmetic circuits (by Neeraj Kayal, see our discussion of lower bounds for arithmetic circuits (above)).
- List decoding Reed–Solomon subcodes up to the singleton bound (by Venkat Guruswami).
- Locally correctable and locally decodable codes over $\mathbb{R}$ with connections to matrix rigidity (by Zeev Dvir).
- List-decoding multivariate multiplicity codes (by Swastik Kopparty, also related to Shubhangi Saraf's plenary presentation).
- The Bourgain–Gamburd–Helfgott approach to analyzing expander graphs (by Amir Yehudayoff, continuing his survey from earlier).
- The De–Mossell–Neeman proof of Borell's Theorem on noise stability in Gaussian space (by Ryan O'Donnell)
- Block-symmetric polynomials correlate with parity better than symmetric (by Emanuele Viola)
- PRGs from shrinkage (by Raghu Meka, in continuation of his plenary presentation).

Workshop: Complexity Theory

Table of Contents

Abstracts

Recent progress in derandomization

Raghu Meka

In this talk I will survey new results related to two important questions in complexity theory: derandomizing constant depth circuits and small space algorithms. We will in particular look at the following results:

(1) PRGs for "garbled" branching programs; cf. [3].
(2) PRGs for combinatorial rectangles; cf. [2].
(3) Deterministic approximate counting for DNFs; cf. [1].

In each case, we will try and highlight new techniques and open problems.

References

[1] Parikshit Gopalan, Raghu Meka, and Omer Reingold. DNF Sparsification and a Faster Deterministic Counting Algorithm. In *IEEE Conference on Computational Complexity*, pages 126–135. IEEE, 2012.

[2] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Better Pseudorandom Generators from Milder Pseudorandom Restrictions. In *the proceedings of the 53rd Annual IEEE Sumposium on Foundations of Comptuer Science, FOCS*. IEEE Computer Society, 2012.

[3] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from Shrinkage. In *the proceedings of the 53rd Annual IEEE Sumposium on Foundations of Comptuer Science, FOCS*. IEEE Computer Society, 2012.

Information and Communication

Anup Rao

Information complexity has played a crucial role in several recent results in computational complexity. In this talk, I focussed on the application of information theory based techniques to proving basic results in communication complexity. I used as a model the recent work in [1], where several direct product results were proved.

Let $\mathsf{suc}(\mu, f, C)$ denote the maximum success probability of a 2-party communication protocol of communication complexity C for computing a function $f(x, y)$ when the inputs are drawn from the distribution μ. Let

$$f^n(x_1, \ldots, x_n, y_1, \ldots, y_n) = (f(x_1, y_1), f(x_2, y_2), \ldots, f(x_n, y_n)),$$

and μ^n denote the product distribution on n pairs of inputs, where each pair is sampled independently according to μ.

In [1], it was proved that:

Theorem 1. *There is a universal constant* $\alpha > 0$ *such that if* $\gamma = 1 - \mathsf{suc}(\mu, f, C)$*,* $T \geq 2$*, and* $T \log^{3/2} T < \alpha\gamma^{5/2} C\sqrt{n}$*, then* $\mathsf{suc}(\mu^n, f^n, T) \leq \exp\left(-\alpha\gamma^2 n\right)$.

I gave an outline of the methods that go into proving such a theorem, which seemed out of reach until very recently.

References

[1] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:143, 2012.

Fully Homomorphic Encryption

Zvika Brakerski

A *Fully Homomorphic Encryption* (FHE) scheme is one that allows to take an encryption of a message $\mathsf{Enc}(m)$ and convert it into $\mathsf{Enc}(f(m))$, for any efficient f, using only public information.

More formally, the key generation process produces a triple of keys (sk, pk, evk), where the public key pk is used by the (randomized) encryption algorithm ($c = \mathsf{Enc}_{pk}(m)$); the secret key sk is used by the decryption algorithm ($m = \mathsf{Dec}_{sk}(c)$); and the *evaluation key* evk which is also public and is used for the purpose of *homomorphic evaluation* $c_f = \mathsf{Eval}_{evk}(f, c_1, \ldots, c_\ell)$ as explained next.

The *correctness* requirement is that homomorphic evaluation indeed transforms an encryption of m into an encryption of $f(m)$, or using the multiple-input notation from the previous paragraph:

$$\mathsf{Dec}_{sk}(c_f) = f(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)) \ .$$

A scheme is *fully* homomorphic if correctness holds for any efficiently computable f. It is not hard to see that it is sufficient to obtain homomorphism with respect to a universal set of gates. This will imply homomorphism using gate-by-gate evaluation. We will use $GF(2)$ as the message space and $\{+, \times\}$ modulo 2 as the universal set of gates.

The *security* requirement is the standard notion of semantic security (CPA security), requiring that there is no efficient distinguisher between the distributions $\mathsf{Enc}_{pk}(m)$ and $\mathsf{Enc}_{pk}(0)$ for any m, even when the distinguisher is given pk and evk.

The concept of FHE was put forth by Rivest, Adleman and Dertouzos [6] back in 1978 but it was not until 2009 when the first candidate scheme was introduced by Gentry [5]. Gentry's scheme, as well as early followups, were based on ideal lattices (or, equivalently, ideals in polynomial rings). This seemed necessary as the ring interpretation allowed to support addition and multiplications simultaneously.

More recently, Brakerski and Vaikuntanathan [4] showed how to obtain FHE without assumptions on ideals. In particular, they showed a scheme that is based on the Learning With Errors (LWE) problem. This scheme was later improved by Brakerski, Gentry and Vaikuntanathan [3] and most recently by Brakerski [2]. We will present a scheme that is inspired by the latter in the spirit of the blog post of Barak and Brakerski [1].

Our scheme is symmetric, namely it requires knowing the secret key for both encryption and decryption, however it can be easily extended to support public key encryption. The basic scheme has a secret key which is just a bit vector

$\mathbf{s} \in \{0,1\}^N$. The ciphertext is a real valued vector $\mathbf{c} \in \mathbb{R}^N$ and the encryption algorithm generates $\mathbf{c}$ such that $\mathbf{s} \cdot \mathbf{c} = m + e + 2I$, where $m \in \{0,1\}$ is the encrypted message, e is a small "noise" value, and $I \in \mathbb{Z}$ is integer. In other words, an encryption of 0 translates to an inner product that is close to an even number and an encryption of 1 translates to an inner product that is close to an odd number. Decryption therefore follows by computing the inner product and rounding. Note that in order to correctly decrypt, it must be that $|e| < 1/2$. The value $|e|$ is referred to as the *noise magnitude* of the ciphertext. The initial noise magnitude in freshly encrypted ciphertexts is a parameter of the scheme and is denoted by α, obviously α cannot be too small or else the scheme becomes insecure, however it is not known how to break the scheme even for $\alpha = 2^{-N^{1/10}}$.

The aforementioned scheme is additively homomorphic by a simple argument, it is clear that $\mathbf{c}_1 + \mathbf{c}_2$ encrypts the message $m_1 + m_2$. The noise magnitude grows by a factor of 2.

For homomorphic multiplication, we use *tensor product*. We observe that $\mathbf{c}_1 \otimes \mathbf{c}_2$ has the property that

$$(\mathbf{c}_1 \otimes \mathbf{c}_2) \cdot (\mathbf{s} \otimes \mathbf{s}) = m_1 m_2 + e' + 2I \ ,$$

for $|e'| \le O(N) \cdot \max\{|e_1|, |e_2|\}$. In some sense, the ciphertext $\mathbf{c}_1 \otimes \mathbf{c}_2$ encrypts the desired $m_1 m_2$, only it has dimension N^2 and it needs a different secret key.

Rewriting $(\mathbf{c}_1 \otimes \mathbf{c}_2) \cdot (\mathbf{s} \otimes \mathbf{s})$ as $\sum_{i,j} (\mathbf{c}_1[i]\mathbf{c}_2[j]) \cdot (\mathbf{s}[i]\mathbf{s}[j])$, we notice that if we provide the appropriate information in the evaluation key, we can bring back the ciphertext to the correct domain. In particular, we will provide in the evaluation key encryptions $\mathbf{x}^{i,j} = \mathsf{Enc}(\mathbf{s}[i]\mathbf{s}[j])$, for all i, j. Given this information, it should hold that

$$\sum_{i,j} (\mathbf{c}_1[i]\mathbf{c}_2[j]) \cdot \mathbf{x}^{i,j}$$

is indeed a valid encryption of $m_1 m_2$ under the key $\mathbf{s}$, which should complete the description of the homomorphic multiplication procedure.

Whereas the aforementioned process only modestly increases the noise magnitude, a more careful examination will reveal that it has a minor flaw that requires correction in the form of providing not only $\mathsf{Enc}(\mathbf{s}[i]\mathbf{s}[j])$ for all i, j, but rather $\mathsf{Enc}(2^{-t}\mathbf{s}[i]\mathbf{s}[j])$ for all i, j, t. We refer to [1] or [2] for a full explanation.

We are left with arguing that providing the aforementioned *evk* does not hurt the security of the scheme. Unfortunately, this cannot be reduced from the security of the scheme itself, but rather requires making an additional *circular security assumption*. The reason is that we are required to provide encryptions of functions of the secret key itself. It is a big open problem to devise an FHE scheme for all functions without making a circular security assumption.

Putting everything together, we showed how to get both additive and multiplicative homomorphism at the cost of an increase in the noise magnitude. A simple calculation shows that the scheme can evaluate circuits of depth roughly $\frac{\log(1/\alpha)}{\log N}$. To obtain a truly fully homomorphic scheme, we use Gentry's *bootstrapping theorem* which argues that so long as the scheme can evaluate circuits deeper than its own *decryption circuit*, it can be made fully homomorphic for any circuit. Since the decryption depth of our scheme is $O(\log N)$, it follows that quasi-polynomially small α is sufficient.

References

[1] B. Barak and Z. Brakerski. The swiss army knife of cryptography. Windows on Theory Blog, 2012. See `http://tiny.cc/fheblog1` and `http://tiny.cc/fheblog2`.

[2] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS*, pages 309–325. ACM, 2012. Invited to ACM Transactions on Computation Theory.

[4] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In R. Ostrovsky, editor, *FOCS*, pages 97–106. IEEE, 2011. Invited to SIAM Journal on Computing.

[5] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[6] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978.

What is the complexity of ensuring differential privacy?

Moritz Hardt

How can we enable useful statistical analyses on a data set while protecting the privacy of those individuals whose data is analyzed? This problem has been studied since the 1970s. The question today is more urgent than ever before due to the increased collection and utility of sensitive data. Differential privacy is a strong notion of privacy protection due to Dwork, McSherry, Nissim and Smith (2006). Intuitively speaking, differential privacy gives the strong guarantee that: The presence or absence of any single individual in a data set will only insignificantly affect the outcome of an analysis. Formally, we say that a randomized algorithm M satisfies *ϵ-differential privacy*, if for every two data sets D and D' differing in only one element and every event S in the output space of the algorithm, we have that

$$\mathbb{P}\left\{M(D) \in S\right\} \leq e^{\epsilon} \cdot \mathbb{P}\left\{M(D') \in S\right\}.$$

Here a data set is just a collection of n data items from some universe which we take to be $\{0,1\}^d$ where d is the dimensionality of the data. The probability above is taken over the randomness of the algorithm. The privacy parameter ϵ is typically thought of as a small constant.

The most basic and well-studied setting of differential privacy is the case where a trusted database curator responds to a number of queries given by an (untrusted) data analyst. The queries that the analyst may ask are so-called *statistical queries*. Statistical queries are a powerful primitive for many tasks that an analyst might want to perform. The answer to a statistical query is a number in $[0,1]$. The goal is to answer a set of statistical queries such that the error on each answer is as small as possible while guaranteeing differential privacy.

High accuracy on huge query sets. In work with Rothblum [1], we give a *privacy-preserving multiplicative weights framework* for this task. At the heart of a our approach is the conceptual insight that private data release can be expressed as a learning problem. The accuracy of our algorithm in terms of database size n and number of queries k nearly matches what is known as the *statistical sampling error* of $O(\sqrt{\log k/n})$. In particular, the algorithm is capable of answering a nearly exponential number of queries with non-trivial accuracy.

The curse of dimensionality. In many applications data is notoriously high-dimensional. Unfortunately, all known differentially private algorithms for answering many queries have a running time that in the worst-case is exponential in the number of dimensions. Can we achieve a polynomial running time?

To address this question, we further develop the connection between privacy and learning theory. Specifically, in work with Rothblum and Servedio [2], we give an efficient reduction from the problem of privately releasing a query class to the problem of non-privately *learning* a function class closely related to the query class. We instantiate this general reduction with a variety of learning algorithms. As a result, we obtain the first *subexponential* time algorithms for privately releasing Boolean conjunctions that are accurate over *any* distribution on conjunctions. Boolean conjunctions form a subclass of statistical queries that has received much attention in differential privacy due to its importance in practice.

Open problems. Two fundamentally different worlds are consistent with our understanding of differential privacy. In one world even simple query classes such as Boolean conjunctions cannot be answered under differential privacy with non-trivial accuracy and a running time that is better than 2^{n^c} for some $c > 0$. In the the other world any simple enough query class can be released in polynomial time and excellent accuracy.

References

[1] Moritz Hardt and Guy Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proc. 51st Foundations of Computer Science (FOCS)*, pages 61–70. IEEE, 2010.

[2] Moritz Hardt, Guy Rothblum, and Rocco Servedio. Private data release via learning thresholds. In *Proc. 23rd Symposium on Discrete Algorithms (SODA)*. ACM-SIAM, 2012.

The Polynomial Freiman-Ruzsa Conjecture in Additive Combinatorics & Applications to Complexity

Noga Ron-Zewi

The polynomial Freiman-Ruzsa conjecture which attempts to classify 'approximte subgroups' of abelian groups is one of the central conjectures in additive combinatorics. When the ambient group is $\mathbb{F}_2^n$, the conjecture is that if $|A + A| \leq K|A|$ then there exists a subset $A' \subseteq A$, $|A'| \geq (1/\text{poly}(K))|A|$ such that $|\text{span}(A')| \leq$

poly$(K)|A'|$, where for a subset $A \subseteq \mathbb{F}_2^n$ we let $A + A = \{a + a' \mid a, a' \in A\}$. In a recent breakthrough, Sanders [San10] managed to prove a quasipolynomial version of this conjecture in which the upper bounds on the ratios $|A|/|A'|$ and $|\mathrm{span}(A')|/|A'|$ are replaced with a bound of the form $K^{O\left(\log^3 K\right)}$.

The complexity theoretic interest in the polynomial Freiman-Ruzsa conjecture started with the works of Samorodnitsky [Sam07], Green and Tao [GT10] and Lovett [Lov10] how showed an application of this conjecture to high-error testing of quadratic polynomials. Later, additional applications were found to the construction of two-source extractors [BZ11], to relating rank to communication complexity [BLR12] and to lower bounds on locally decodable matching vector codes [BDL12]. All latter applications are derived via the approximate duality conjecture, introduced by Ben-Sasson and Ron-Zewi [BZ11], which was shown to have tight relations with the polynomial Freiman-Ruzsa conjecture.

In the talk I will introduce the polynomial Freiman-Ruzsa and approximate duality conjectures and survey the relations between them as well as their applications to complexity theory.

References

[BDL12] Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New bounds for matching vector codes. 2012. Preprint.

[BLR12] Eli Ben-Sasson, Shachar Lovett, and Noga Ron-Zewi. An additive combinatorics approach relating rank to communication complexity. In *the proceedings of the 53rd Annual IEEE Sumposium on Foundations of Comptuer Science, FOCS*. IEEE Computer Society, 2012.

[BZ11] Eli Ben-Sasson and Noga Zewi. From affine to two-source extractors via approximate duality. In *the Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, STOC*. ACM Press, 2011.

[GT10] Ben Green and Terence Tao. An equivalence between inverse sumset theorems and inverse conjectures for the u^3 norm. *Math. Proc. Cambridge Philos. Soc.*, 149(1):1–19, 2010.

[Lov10] Shachar Lovett. Equivalence of polynomial conjectures in additive combinatorics. *To appear, Combinatorica*, 2010.

[Sam07] Alex Samorodnitsky. Low-degree tests at large distances. In *the Proceedings of the 39th Annual ACM Symposium on Theory of Computing, STOC*, pages 506–515. ACM Press, 2007.

[San10] Tom Sanders. On the Bogolyubov-Ruzsa lemma. *To appear, Anal. PDE*, 2010.

On the Real τ-Conjecture — An Approach to Permanent Lower Bounds

Pascal Koiran

According to the real τ-conjecture [5], the number of real roots of a sum of products of sparse polynomials should be polynomially bounded in the size of such an expression. By contrast, the original τ-conjecture of Shub and Smale[8] deals with integer roots of arbitrary straight-line programs, and is known to become false for real rather than integer roots. Both conjectures imply that the permanent is hard to compute for arithmetic circuits. In this talk, I sketched the proof of this implication for the real τ-conjecture. The two main ingredients are:

(i) Reduction to depth 4 for arithmetic circuits [1, 6].

(ii) A connection between the counting hierarchy and arithmetic circuit complexity discovered in a paper by Allender et al. [2] and further explored by Bürgisser [3].

I also discussed a tractable case of the conjecture which leads to unconditional lower bounds and polynomial identity testing in a restricted model [4, 7].

References

[1] M. Agrawal and V. Vinay. Arithmetic circuits: a chasm at depth four. In *Proc. 49th IEEE Symposium on Foundations of Computer Science*, 2008.

[2] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. Bro-Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009. Conference version in CCC 2006.

[3] P. Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18:81–103, 2009. Conference version in STACS 2007.

[4] B. Grenet, P. Koiran, N. Portier, and Y. Strozecki. The limited power of powering: polynomial identity testing and a depth-four lower bound for the permanent. In *Proc. FSTTCS*. Springer, 2011.

[5] P. Koiran. Shallow circuits with high-powered inputs. In *Proc. Second Symposium on Innovations in Computer Science (ICS 2011)*, 2011.

[6] P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Thoretical Computer Science*, 448:56–65, 2012.

[7] P. Koiran, N. Portier, and S. Tavenas. A wronskian approach to the real τ-conjecture. 2012.

[8] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP". *Duke Mathematical Journal*, 81(1):47–54, 1995.

Multiplicity Codes

SHUBHANGI SARAF

(joint work with Swastik Kopparty, Sergey Yekhanin)

Classical error-correcting codes allow one to encode a k-bit message $\mathbf{x}$ into an n-bit codeword $C(\mathbf{x})$, in such a way that $\mathbf{x}$ can still be recovered even if $C(\mathbf{x})$ gets corrupted in a number of coordinates. The traditional way to recover information about $\mathbf{x}$ given access to a corrupted version of $C(\mathbf{x})$ is to run a decoder for C, which would read and process the entire corrupted codeword, and then recover the entire original message $\mathbf{x}$. Suppose that one is only interested in recovering a single bit or a few bits of $\mathbf{x}$. In this case, codes with more efficient decoding schemes are possible, allowing one to read only a small number of code positions. Such codes are known as Locally Decodable Codes (LDCs). Locally decodable codes allow reconstruction of an arbitrary bit $\mathbf{x}_i$, by looking only at $t \ll k$ randomly chosen coordinates of (a possibly corrupted) $C(\mathbf{x})$.

The main parameters of a locally decodable code that measure its utility are the codeword length n (as a function of the message length k) and the query complexity of local decoding. The length measures the amount of redundancy that is introduced into the message by the encoder. The query complexity counts the number of bits that need to be read from a (corrupted) codeword in order to recover a single bit of the message. Ideally, one would like to have both of these parameters as small as possible. One however cannot minimize the codeword length and the query complexity simultaneously; there is a trade-off. On one end of the spectrum we have LDCs with the codeword length close to the message length, decodable with somewhat large query complexity. Such codes are useful for data storage and transmission. On the other end we have LDCs where the query complexity is a small constant but the codeword length is large compared to the message length. Such codes find applications in complexity theory and cryptography. The true shape of the trade-off between the codeword length and the query complexity of LDCs is not known. Determining it is a major open problem (see [Yek10] for a recent survey of the LDC literature).

While most prior work focuses on the low query (and even constant query) regime, in this work we will look at the other extreme and consider the setting of locally decodable codes with very low redundancy, which may be of even greater practical interest. More precisely, we will be interested in minimizing the query complexity of local decoding for codes of large *rate* (defined as the ratio k/n, where the code encodes k bits into n bits). For codes of rate $> 1/2$, it was unknown how to get any nontrivial local decoding whatsoever. For smaller rates, it was known how to construct codes (in fact, the classical Reed-Muller codes based on evaluating multivariate polynomials have this property) which admit local decoding with $O(k^{\epsilon})$ queries and time, at the cost of reducing the rate to $\epsilon^{\Omega(1/\epsilon)}$.

In this paper, we introduce a new and natural family of locally decodable codes, which achieve high rates while admitting local decoding with low query complexity. These codes, which we call multiplicity codes, are based on evaluating multivariate polynomials and their derivatives. They inherit the local-decodability of the traditional multivariate polynomial codes, while achieving better tradeoffs and flexibility in the rate and minimum distance. Using multiplicity codes, we prove that it is possible to have codes that simultaneously have (a) rate approaching 1, and (b) allow for local decoding with arbitrary polynomially-small time and query complexity.

Main Theorem (informal): *For every $\epsilon > 0, \alpha > 0$, and for infinitely many k, there exists a code which encodes k-bit messages with rate $1 - \alpha$, and is locally decodable from some constant fraction of errors using $O(k^{\epsilon})$ time and queries.*

Previous work on locally decodable codes

Locally decodable codes have been implicitly studied in coding theory for a very long time, starting with Reed's "majority-logic decoder" for binary Reed-Muller codes [Ree54]. In theoretical computer science, locally decodable codes (and in particular, locally decodable codes based on multivariate polynomials) have played an important part in the Proof-Checking Revolution of the early 90s, as well as in other fundamental results in complexity theory. Locally decodable codes were first formally defined by Katz and Trevisan [KT00]. Since then, the quest for understanding locally decodable codes has generated many developments.

Most of the previous work on LDCs has focussed on local decoding with a constant number of queries. For a long time, it was generally believed that for decoding with constantly many queries, a k bit message must be encoded into at least $\exp(k^\alpha)$ bits, for constant $\alpha > 0$. Recently, in a surprising sequence of works [Yek08, Efr09] this was shown to be soundly false; today we know constant query locally decodable codes which encode k bits into as few as $\exp(\exp(\log^\alpha(k)))$ bits for constant $\alpha > 0$.

There has also been considerable work on the problem of proving lower bounds on the length of locally decodable codes. In particular, it is known [KT00] that for codes of constant rate, local decoding requires at least $\Omega(\log k)$ queries. For codes locally decodable with $\omega(\log k)$ queries, no nontrivial lower bound on the length on the code is known. For error-correction with $O(k^\epsilon)$ queries, Dvir [Dvi10] recently conjectured a lower bound on the length of some closely related objects called *locally self-correctable codes*. Precisely, the conjecture states that for every field $\mathbb{F}$, there exist positive constants α and ϵ such that there are no linear codes over $\mathbb{F}$ of length n, rate $1 - \alpha$ and locally self-correctable with query complexity $O(n^\epsilon)$ from a certain sub-constant fraction of errors. Dvir [Dvi10] then showed that establishing this conjecture would yield progress on some well-known open questions in arithmetic circuit complexity. Our results refute Dvir's conjecture over finite fields.

Applications of derivatives and multiplicities: The notions of derivative and multiplicity have played an important role in several prior works in coding theory and theoretical computer science. The "method of multiplicities" is a powerful combinatorial/algorithmic technique which has been developed and used in a number of contexts in recent years [GS99, PV05, GR08, DKSS09]. It is a method for analyzing subsets of $\mathbb{F}_q^m$ by interpolating a polynomial that vanishes at each point of that subset with high multiplicity; this often yields a strengthening of the "polynomial method", which would analyze such a subset by interpolating a polynomial that simply vanishes at each point of that subset. Xing [Xin03] considers the space of differentials on an algebraic curve to prove the existence of error-correcting codes above the Tsfasman-Vladut-Zink bound. Woodruff and Yekhanin [WY05] use evaluations of polynomials and their derivatives to construct private information retrieval schemes with improved communication complexity. Multiplicity codes add to this body of work, which follows the general theme that wherever polynomials and their zeroes are useful, also considering their derivatives and high-multiplicity zeroes can be even more useful.

References

[DKSS09] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. In *50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 181–190, 2009.

[Dvi10] Zeev Dvir. On matrix rigidity and locally self-correctable codes. In *26th IEEE Computational Complexity Conference (CCC)*, pages 102–113, 2010.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *41st ACM Symposium on Theory of Computing (STOC)*, pages 39–44, 2009.

[GKST02] Oded Goldreich, Howard Karloff, Leonard Schulman, and Luca Trevisan. Lower bounds for locally decodable codes and private information retrieval. In *17th IEEE Computational Complexity Conference (CCC)*, pages 175–183, 2002.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *32nd ACM Symposium on Theory of Computing (STOC)*, pages 80–86, 2000.

[PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *46th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005.

[Ree54] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

[WY05] Woodruff and Sergey Yekhanin. A geometric approach to information theoretic private information retrieval. In *20th IEEE Computational Complexity Conference (CCC)*, pages 275–284, 2005.

[Xin03] Chaoping Xing. Nonlinear codes from algebraic curves improving the Tsfasman-Vladut-Zink bound. *IEEE Transactions on Information Theory*, 49(7):1653–1657, 2003.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM*, 55:1–16, 2008.

[Yek10] Sergey Yekhanin. Locally decodable codes. *Foundations and trends in theoretical computer science*, 2010. to appear.

Size Lowerbounds for Mathematical Programs

David Steurer

Most combinatorial optimization problems can be formulated as maximizing a *linear objective function* over a (finite) set of points, called *solutions*, such that the linear function encodes the instance of the problem and the set of solutions depends only on the size of the instance (but not on other characteristics of the instance). For example in the traveling salesperson problem on n cities (TSP_n), solutions can be 0/1 vectors $x \in \mathbb{R}^{n^2}$ corresponding to tours of the cities, and instances can be encoded as linear functions $\sum_{i,j\in[n]} d_{ij}x_{ij}$ on $\mathbb{R}^{n^2}$.

Given such an encoding of a combinatorial optimization problem Π_n,[1] we can ask if there exists a linear or semidefinite program $\mathcal{P}_n$ of small size such that (1) every solution of the problem Π_n is a feasible solution for the mathematical program $\mathcal{P}_n$ (i.e., satisfying all constraints of the program) and (2) solving the program $\mathcal{P}_n$ on a linear objective function corresponding to an instance of Π_n yields a value equal (or close to) the optimal value of the instance. This question is interesting because for many basic optimization problems, e.g., TSP and the maximum cut problem (Max Cut), the above *mathematical relaxation* approach captures the best known algorithms.

[1] The subscript n indicates that we restrict the problem to instances of size n (e.g., TSP instances on n cities)

Yannakakis [Yan91] first formulated the above question and provided a characterization of the minimum size of linear programs in terms of non-negative factorizations and communication complexity. He also showed an exponential lowerbound on the size of *symmetric* linear programs for TSP_n.

Fiorini et al. [FMP+12] first obtained lowerbounds on the size of general (non-symmetric) linear programs. Based on a linear lowerbound on the nondeterministic communication complexity of the unique disjoint problem[2] [KS92, Raz92, dW03], they showed exponential lowerbounds on the size of (non-symmetric) linear programs for TSP_n and the independent set problem ($\textsc{Ind Set}_n$).

All of the works above are about linear programs that compute exact solutions for a given optimization problem. In the context of optimization, it is natural to ask about linear program that provide approximate solutions for the problem. Braun et al. [BFPS12] first addressed this question and showed that an approximation factor of $n^{1/2-\varepsilon}$ for $\textsc{Ind Set}_n$ requires linear programs of size $2^{n^{\Omega(\varepsilon)}}$. This result is based on quantitative refinements of Razborov's rectangle corruption lemma for disjointness [Raz92]. Braverman and Moitra [BM12] obtained a quantitatively optimal lowerbound, ruling out linear programs of size $2^{n^{o(\varepsilon)}}$ that achieve an approximation factor of $n^{1-\varepsilon}$ for $\textsc{Ind Set}_n$. This result is based on a strengthening of the information-theoretic lowerbounds for the communication complexity of unique disjointness [BYJKS04].

We highlight two outstanding open problems in this context: (1) For some constraint satisfaction problem, rule out that there exists a linear programs of size $\mathrm{poly}(n)$ that achieves an approximation factor of 0.999 for all instances on n variables. This result would be an analog of the PCP theorem and could imply many other approximation lower bounds via gadget reductions. (2) For some combinatorial optimization problem, rule out that there exists a semidefinite program of size $\mathrm{poly}(n)$ that solves all instances of size n of the problem (without approximation). Currently, no non-trivial lower bounds for semidefinite programs are known (even symmetric ones).

References

[BFPS12] Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer, *Approximation limits of linear programs (beyond hierarchies)*, FOCS, 2012, pp. 480–489.

[BM12] Mark Braverman and Ankur Moitra, *An information complexity approach to extended formulations*, Electronic Colloquium on Computational Complexity (ECCC) **19** (2012), 131.

[BYJKS04] Z. Bar-Yossef, T.S. Jayram, R. Kumar, and D. Sivakumar, *An information statistics approach to data stream and communication complexity*, J. Comput. System Sci. **68** (2004), no. 4, 702–732.

[dW03] Ronald de Wolf, *Nondeterministic quantum query and communication complexities*, SIAM J. Comput. **32** (2003), no. 3, 681–699.

[FMP+12] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf, *Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds*, STOC, 2012, pp. 95–106.

[2] In this promise communication problem, two parties aim to distinguish between the case that their input are disjoint sets and the case that their input sets have a unique intersection.

[KS92] B. Kalyanasundaram and G. Schnitger, *The probabilistic communication complexity of set intersection*, SIAM J. Discrete Math. **5** (1992), 545–557.

[Raz92] A. A. Razborov, *On the distributional complexity of disjointness*, Theoret. Comput. Sci. **106** (1992), no. 2, 385–390.

[Yan91] Mihalis Yannakakis, *Expressing combinatorial optimization problems by linear programs*, J. Comput. Syst. Sci. **43** (1991), no. 3, 441–466.

Update on the status of the Unique Games Conjecture

Boaz Barak

Khot's Unique Games conjecture [4] has been the center of much exciting research in recent years. This talk surveyed the currents status of the efforts to understand whether or not the conjecture is true. An emerging theme in this research is the question of the power of *semi-definite programming* in the context of optimization problem. On one hand, if the conjecture is true, then the basic semidefinite program for many optimization problems is optimal [5], in the sense that beating it would be NP-hard. On the other hand, extensions of this semidefinite program provide the best candidates to refute this conjecture.

We discussed the currently known best algorithms for the unique games computational problem, including a subexponential algorithm for all instances [1] and a polynomial-time algorithm for some interesting instances [2], as well as the best candidates for instances that are "hard" for at least some classes of algorithm [3]. We presented a phenomena which underlies the difficulty in coming up with candidate hard instances for unique games: one can often transform the proof that the instance I has a certain property P into a proof that the "Sum of Squares" semidefinite programming hierarchy can efficiently certify that I has P.

References

[1] S. Arora, B. Barak, and D. Steurer. *Subexponential Algorithms for Unique Games and Related Problems.* In *FOCS*, pages 563–572, 2010.

[2] B. Barak, F. G. S. L. Brandão, A. W. Harrow, J. A. Kelner, D. Steurer, and Y. Zhou. *Hypercontractivity, sum-of-squares proofs, and their applications.* In *STOC*, pages 307–326, 2012.

[3] B. Barak, P. Gopalan, J. Håstad, R. Meka, P. Raghavendra, and D. Steurer. *Making the long code shorter.* In *FOCS*, 2012.

[4] S. Khot. *On the power of unique 2-prover 1-round games.* In *STOC*, pages 767–775, 2002.

[5] P. Raghavendra. *Optimal algorithms and inapproximability results for every CSP?* In *STOC*, pages 245–254, 2008.

Cryptographic Hardness of Random Local Functions – Survey

Benny Applebaum

Constant parallel-time cryptography allows performing complex cryptographic tasks at an ultimate level of parallelism, namely, by local functions that each of their output bits depend on a constant number of input bits. The feasibility of such highly efficient cryptographic constructions was widely studied in the last decade via two main research threads.

The first is an encoding-based approach, developed in [1, 2], in which standard cryptographic computations are transformed into local computations via the use of special encoding schemes called *randomized encoding* of functions. The second approach, initiated by Goldreich [3], is more direct and it conjectures that almost all non-trivial local functions have some cryptographic properties.

In this survey we focus on the latter approach. We consider *random local functions* in which each output bit is computed by applying some fixed d-local predicate P to a randomly chosen d-size subset of the input bits. Formally, this can be viewed as selecting a random member from a collection $\mathcal{F}_{P,n,m}$ of d-local functions where each member $f_{G,P} : \{0,1\}^n \rightarrow \{0,1\}^m$ is specified by a d-uniform hypergraph G with n nodes and m hyperedges, and the i-th output of $f_{G,Q}$ is computed by applying the predicate P to the d inputs that are indexed by the i-th hyperedge.

We survey several basic issues regarding the cryptographic hardness of random local functions. These include known attacks, hardness against restricted algorithms, pseudorandomness, collision resistance, and connections to other problems in computational complexity and cryptography. We also present some open questions with the hope to develop a systematic study of the cryptographic hardness of local functions, which will eventually lead to a comprehensive theory of "locally-computable" cryptography.

References

[1] B. Applebaum, Y. Ishai, and E. Kushilevitz, *Cryptography in* NC^0, SIAM Journal on Computing, **36(4)** (2006), 845–888.

[2] B. Applebaum, Y. Ishai, and E. Kushilevitz, *Computationally private randomizing polynomials and their applications*, Journal of Computational Complexity, **15(2)** (2006), 115–162.

[3] O. Goldreich, *Candidate one-way functions based on expander graphs*, Electronic Colloquium on Computational Complexity (ECCC), **7(090)** (2000).

On the recent progress on matrix multiplication

Virginia Vassilevska Williams

The product of two matrices is one of the most basic operations in mathematics and computer science. Many other essential matrix operations can be efficiently reduced to it, such as Gaussian elimination, LUP decomposition, the determinant or the inverse of a matrix. Matrix multiplication is also used as a subroutine in many computational problems that, on the face of it, have nothing to do with matrices, e.g. graph transitive closure and context free grammar parsing.

Until the late 1960s it was believed that computing the product C of two $n \times n$ matrices requires essentially a cubic number of operations, as the fastest algorithm known was the naive algorithm which indeed runs in $O(n^3)$ time. In 1969, Strassen [8] excited the research community by giving the first subcubic time algorithm for matrix multiplication, running in $O(n^{2.808})$ time. This amazing discovery spawned a long line of research which gradually reduced the matrix multiplication exponent ω over time. In 1978, Pan [4] showed $\omega < 2.796$. The following year, Bini et al. [1] introduced the notion of *border rank* and obtained $\omega < 2.78$. Schönhage [6] generalized this notion in 1981, proved his τ-theorem (also called the asymptotic sum inequality), and showed that $\omega < 2.548$. In the same paper, combining his work with ideas by Pan, he also showed $\omega < 2.522$. The following year, Romani [5] found that $\omega < 2.517$. The first result to break 2.5 was by Coppersmith and Winograd [2] who obtained $\omega < 2.496$. In 1986, Strassen [9] introduced his *laser* method which allowed for an entirely new attack on the matrix multiplication problem. He also decreased the bound to $\omega < 2.479$. Three years later, Coppersmith and Winograd [3] combined Strassen's technique with a novel form of analysis based on large sets avoiding arithmetic progressions and obtained the famous bound of $\omega < 2.376$ which remained unchanged for more than twenty years. The bound was recently improved by Stothers [7] and myself [10]. The purpose of this talk is to highlight these improvements and the work leading up to them.

The basic idea of all approaches for matrix multiplication since 1981, including Coppersmith-Winograd (CW), is as follows. One first constructs an algorithm A which given Q-length vectors x and y for constant Q, computes Q values of the form $z_k = \sum_{i,j} t_{ijk} x_i y_j$, say with $t_{ijk} \in \{0, 1\}$, using a smaller number of products than would naively be necessary. The values z_k do not necessarily have to correspond to entries from a matrix product. Then, one considers the algorithm A^n obtained by applying A to vectors x, y of length Q^n, recursively n times as follows. Split x and y into Q subvectors of length Q^{n-1}. Then run A on x and y treating them as vectors of length Q with entries that are vectors of length Q^{n-1}. When the product of two entries is needed, use A^{n-1} to compute it. This algorithm A^n is called the *nth tensor power* of A. Its running time is essentially $O(r^n)$ if r is the number of multiplications performed by A.

The goal of the approach is to show that for very large n one can set enough variables x_i, y_j, z_k to 0 so that running A^n on the resulting vectors x and y actually computes a matrix product. That is, as n grows, some subvectors x' of x and y' of y can be thought to represent square matrices and when A^n is run on x and y, a subvector of z is actually the matrix product of x' and y'.

If A^n can be used to multiply $m \times m$ matrices in $O(r^n)$ time, then this implies that $\omega \leq \log_m r^n$, so that the larger m is, the better the bound on ω.

Coppersmith and Winograd [3] introduced techniques which, when combined with previous techniques by Schönhage [6] and Strassen [9], allowed them to effectively choose which variables to set to 0 so that one can compute very large matrix products using A^n. Part of their techniques rely on partitioning the index triples $i, j, k \in [Q]^n$ into groups and analyzing how "similar" each group g computation $\{z_{kg} = \sum_{i,j:\ (i,j,k)\in g} t_{ijk} x_i y_j\}_k$ is to a matrix product. The similarity measure used is called the *value* of the group.

Depending on the underlying algorithm A, the partitioning into groups varies and can affect the final bound on ω. Coppersmith and Winograd analyzed a particular algorithm A which resulted in $\omega < 2.39$. Then they noticed that if one uses A^2 as the basic algorithm (the "base case") instead, one can obtain the better bound $\omega < 2.376$. They left as an open problem what happens if one uses A^3 as the basic algorithm instead.

Many people attempted to analyze the third tensor power (from personal communication with Umans, Kleinberg and Coppersmith), and found the result to be very disappointing. In fact no improvement whatsoever on 2.376 can be found! This finding led some to believe that 2.376 may be the final answer, at least for the CW algorithm. Furthermore, with each new tensor power, the number of new values that need to be analyzed grows quadratically. For the eighth tensor power for instance, 30 separate analyses are required! Seemingly, each of these analyses requires a separate application of the CW techniques, and hence analyzing larger tensor powers would seem to require an enormous amount of patience. Since the third tensor power does not give any improvement, the prospects looked bleak. The approach was abandoned for more than 20 years.

The first improvement over Coppersmith-Winograd was obtained by Stothers in his thesis [7]. There, he argues that $\omega < 2.3737$ by analyzing the 4th tensor power of the Coppersmith-Winograd construction. To obtain his improvement, he analyzes 10 different values, all by hand. Stothers, however, uses a shortcut in his analysis that allows him to formulate the values of the groups of the fourth tensor power in terms of the values of groups of the second tensor power. This shortcut simplifies much of the work needed.

My work [10], done largely independently from Stothers, gives a new general framework to tightly analyze the techniques behind the CW approach [3] *entirely by computer*. (Thus the cumbersome part of analyzing values by hand is eliminated.) My paper demonstrates the effectiveness of the new analysis by showing that the 8th tensor power of the CW algorithm [3] in fact gives $\omega < 2.3727$. (It is likely that higher tensor powers can give tighter estimates, and this could be the subject of future work.)

There are two main theorems behind my approach. The first theorem takes any tensor power A^n of a basic algorithm A, picks a particular group partitioning for A^n and derives an efficient procedure computing formulas for the values of these groups. The second theorem assumes that one knows the values for A^n and derives an efficient procedure which outputs a (nonlinear) constraint program on $O(n^2)$ variables, the solution of which gives a bound on ω. The two procedures combined give an algorithm which searches for matrix multiplication algorithms, thus also looking for a bound on ω.

The algorithm computing bounds on the group values boils down to solving linear systems of equations and linear programs, and the algorithm formulating the nonlinear program defining the search space for matrix multiplication algorithms boils down to just solving linear systems. As linear systems can be solved using matrix multiplication algorithms, we get the following curious phenomenon.

Good matrix multiplication algorithms can be used to prove theorems about the existence of better matrix multiplication algorithms.

In my paper, I apply the procedures given by the theorems to the second, third, fourth and eighth tensor powers of the Coppersmith-Winograd algorithm, obtaining improved bounds with each new tensor power (except the third). To minimize the computational overhead, after seeing Stothers' work, I incorporated Stothers' shortcut into my value theorem. This made analyzing tensor powers that are powers of two particularly cheap computationally and led to the analysis of the 8th tensor power.

Similar to [3], my new proofs apply to any starting algorithm that satisfies a simple uniformity requirement. The upshot of the approach is that now any such algorithm and its higher tensor powers can be analyzed entirely by computer. (In fact, our analysis of the 8th tensor power of the CW algorithm is done this way.) The burden is now entirely offloaded to constructing base algorithms satisfying the requirement.

References

[1] D. Bini, M. Capovani, F. Romani, and G. Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979.

[2] D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. In *Proc. SFCS*, pages 82–90, 1981.

[3] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computation*, 9(3):251–280, 1990.

[4] V. Y. Pan. Strassen's algorithm is not optimal. In *Proc. FOCS*, volume 19, pages 166–176, 1978.

[5] F. Romani. Some properties of disjoint sums of tensors related to matrix multiplication. *SIAM J. Comput.*, pages 263–267, 1982.

[6] A. Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981.

[7] A. Stothers. *Ph.D. Thesis, U. Edinburgh*, 2010.

[8] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[9] V. Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *FOCS*, pages 49–54, 1986.

[10] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proc. STOC*, pages 887–898, 2012.

Recent progress on matrix multiplication II: potential routes to $\omega = 2$

Chris Umans
(joint work with Noga Alon, Henry Cohn, Amir Shpilka)

As usual, ω is exponent of matrix multiplication (over $\mathbb{C}$). We assume the reader is familiar with the basic definitions of tensors, tensor rank, denoted $R(\cdot)$, and border rank. As is standard, we denote by $\langle n, m, p\rangle$ the tensor associated with $n \times m$ by $m \times p$ matrix multiplication.

1. Conjectures implying $\omega = 2$. We begin be recalling several concrete conjectures that would imply $\omega = 2$. The first two are due to Coppersmith and Winograd [6], and the second two are due to Cohn, Kleinberg, Szegedy and Umans [3].

We write tensors as formal trilinear forms. The family of tensors $T_q = \sum_{i=1}^{q} X_0Y_iZ_i + X_iY_0Z_i + X_iY_iZ_0$ were used by Coppersmith and Winograd to achieve $\omega < 2.41$ in a "warm-up" to their well-known 1990 result. It is known that the border rank of T_q is $q + 2$. If the *asymptotic rank* of the tensor T_2 is 3, then $\omega = 2$.

Conjecture 1 (asymptotic rank[6])**.** *As* $n \to \infty$, *we have* $R(T_2^{\otimes n})^{1/n} \to 3$.

The tensor $\sum_{i,j,k\in\{0,1,2\},i+j+k=0 \bmod 3} X_iY_jZ_k$ is similar to T_2 and has rank 3. The next conjecture would enable this tensor to be used in a straightforward manner (see [2]) to obtain $\omega = 2$. A subset S of an abelian group A is said to contain three *disjoint equivoluminous subsets* if there exist disjoint non-empty subsets $X, Y, Z \subseteq S$ for which $\sum_{x\in X} x = \sum_{y\in Y} y = \sum_{z\in Z} z$.

Conjecture 2 (no three equivoluminous subsets [6])**.** *There exist finite abelian groups* A_i *with subsets* $S_i \subseteq A_i$ *containing no three disjoint equivoluminous subsets, and* $|A_i| \leq 2^{o(|S_i|)}$.

The next two conjectures arose in the "group-theoretic" framework suggested by Cohn and Umans [4]. A *uniquely solvable puzzle* is[3] a collection of partitions A_i, B_i, C_i of $[n]$, with the property that for all i, j, k not all equal, there is some x in 2 or 3 of the sets A_i, B_j, C_k. It is not hard to see that the cardinality of a uniquely solvable puzzle can be at most $\binom{n}{n/3}^{1+o(1)}$, and it follows from [6] that cardinality $\binom{n}{n/3}^{1-o(1)}$ can be achieved. A *strong uniquely solvable puzzle* replaces the condition "there is some x in 2 or 3 of the sets A_i, B_j, C_k" with the stronger condition "there is some x in exactly 2 of the sets A_i, B_j, C_k". If strong uniquely solvable puzzles exist with cardinality similar to that achievable by (non-strong) uniquely solvable puzzles, then $\omega = 2$.

Conjecture 3 (strong uniquely solvable puzzles [3])**.** *There exist strong uniquely solvable puzzles of cardinality* $\binom{n}{n/3}^{1-o(1)}$.

Also from [3], we have the following conjecture, which would also imply $\omega = 2$.

[3] These definitions are of the so-called "local" variant of the object.

Conjecture 4 (two families [3])**.** *There exist abelian groups H and subsets $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$ of H with $|A_i| \cdot |B_i| \geq n^{2-o(1)}$ for all i and $|H| \leq n^{2+o(1)}$, such that (1) $|A_i + B_i| = |A_i| \cdot |B_i| \; \forall i$, and (2) $(A_i + B_i) \cap (A_j + B_k) = \emptyset \; \forall i, j \neq k$.*

Alon, Shpilka, and Umans [1] made progress on understanding Conjectures 2 and 3 by relating them to sunflower conjectures. Recall that sets $S_1, S_2, \ldots S_k$ form a *k-sunflower* if their pairwise intersections are all equal. The following is the famous Erdös-Rado sunflower conjecture (specialized to 3-sunflowers):

Conjecture 5 (classical sunflower [1])**.** *There exists a constant c such that every collection of s-subsets having cardinality at least c^s contains a 3-sunflower.*

This conjecture is well-known and widely believed to be true; hence the following theorem suggests that Conjecture 2 is unlikely to be true.

Theorem 6 ([1])**.** *If Conjecture 2 is true, then Conjecture 5 is false.*

Another well-known question in combinatorics (that can be interpreted in terms of sunflowers – see [1]) is how large a subset of $S \subseteq \mathbb{Z}_3^n$ can one have such that for all $x, y, z \in S$, $x+y+z = 0 \Rightarrow x = y = z$? Some suspect that cardinality $3^{n(1-o(1))}$ is achievable; on the other hand, one might conjecture that there is a universal constant $c > 0$ such that $3^{n(1-c)}$ is the best achievable. A stronger conjecture is that the same is true for a "multicolored" version:

Conjecture 7 (multicolored sunflowers in $\mathbb{Z}_3^n$ [1])**.** *There exists a constant $c > 0$ for which every subset of triples $S \subseteq (\mathbb{Z}_3^n)^3$ having cardinality at least $3^{n(1-c)}$ and for which every $(x, y, z) \in S$ satisfies $x + y + z = 0$, contains three elements $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$ for which $x_1 + y_2 + z_3 = 0$.*

There is no strong consensus on whether this conjecture is likely to be true or false; however the following theorem suggests that understanding it is a prerequisite to understanding Conjecture 3:

Theorem 8 ([1])**.** *If Conjecture 3 is true, then Conjecture 7 is false.*

Note that while these theorems suggest that Conjectures 2 and 3 may be difficult and/or false, Conjectures 1 and 4 remain viable routes to proving $\omega = 2$.

2. The embedding approach. We now turn to discussing an approach suggested by Cohn and Umans [4], developed by Cohn et al. [3], and recently generalized by Cohn and Umans [5]. The general idea is to obtain bounds on the rank of the matrix multiplication tensor by embedding it into the $\mathcal{A}$-multiplication tensor for a suitable semi-simple algebra $\mathcal{A}$. Since the semi-simple algebra is isomorphic to block-diagonal matrix multiplication (where the block sizes are the dimensions of the irreducible representations), the overall effect is to reduce a single large matrix multiplication to several smaller matrix multiplications. In such a case we say that $\mathcal{A}$ *realizes* matrix multiplication. Depending on the relative sizes, this can imply strong bounds on ω.

For this to be an effective strategy, the algebra should have a "nice" basis which allows one to express sufficient conditions for the existence of such an embedding in terms of an underlying combinatorial or algebraic object. For example, if $\mathcal{A}$ is the group algebra $\mathbb{C}[G]$, the group elements are the "nice basis", and a sufficient condition for realizing $\langle n, m, p\rangle$ is for G to have three subgroups X, Y, Z that satisfy the *triple product property*[4]: $xyz = 1 \Leftrightarrow x = y = z = 1$, (with $x \in X, y \in Y, z \in Z$ and $|X| = n$, $|Y| = m$, $|Z| = p$). Note that this condition makes no direct reference to the algebra $\mathbb{C}[G]$ and is expressed entirely in terms of the underlying group G.

Using this "group-theoretic approach" (as developed in [4, 3]) we obtain the following theorem

Theorem 9 ([4])**.** *If a group G with irreducible representations of dimensions $d_1, \dots, d_k$ realizes $\langle n, n, n\rangle$, then $n^\omega \le \sum_{i=1}^k d_i^\omega$.*

The right hand side is always upper-bounded by $d_{\max}^{\omega-2}|G|$, and this is frequently very close to being tight. Then one can see that to obtain $\omega = 2$ within this framework, one needs a group G of size $n^{2+o(1)}$ *and* with $d_{\max}$ "small" (certainly $d_{\max} = n^{o(1)}$ suffices). It is our experience that there is a tension between a group G realizing $\langle n, n, n\rangle$ for n large, and $d_{\max}$ being small. The generalization we will discuss next obviates the need to think about $d_{\max}$ (and the representation theory of the algebra in general), because it can work entirely in the commutative setting. In contrast, it is an easy exercise to show that non-abelian groups are necessary to prove non-trivial bounds on ω in the group-theoretic setting.

Recently, we have generalized the approach based on embedding into group algebras to general semi-simple algebras [5]. Such an algebra $\mathcal{A}$ is specified by a basis $e_1, e_2, \dots, e_k$, together with structure constants $\lambda_{i,j,k}$ for which $e_ie_j = \sum_k \lambda_{i,j,k}e_k$. The tensor associated with $\mathcal{A}$-multiplication is $\sum_{i,j,k} \lambda_{i,j,k}X_iY_jZ_k$, and we wish to find a matrix multiplication tensor within it. Here a complication arises: while for group algebras, the $\lambda_{i,j,k}$ are all zero or one, this is not generally true for most interesting semi-simple algebras. So it seems most natural to find a tensor *with the same support as a matrix multiplication tensor* within the $\mathcal{A}$-algebra multiplication tensor. The rank of such a tensor is then bounded by the rank of the $\mathcal{A}$-algebra multiplication tensor, which is well-understood (in principle) because $\mathcal{A}$-multiplication is isomorphic to block-diagonal matrix multiplication. But is a rank bound on a tensor with the same support as the matrix multiplication tensor useful for bounding ω? We answer this question next.

Define the *s-rank* of a tensor T, denoted $R_s(T)$, to be the minimum rank of a tensor having the same support as T. The concept of s-rank seems interesting in its own right; we know of examples where s-rank can be much smaller than rank (and border-rank), and also examples where border rank is smaller than s-rank.

We define ω_s in analogy with ω (whose definition is below for comparison):

$$\begin{aligned}\omega &= \inf\{\tau : R(\langle n, n, n\rangle) \le O(n^\tau)\}\\ \omega_s &= \inf\{\tau : R_s(\langle n, n, n\rangle) \le O(n^\tau)\}.\end{aligned}$$

[4] In fact one should allow X, Y, Z to be subsets and then x, y, z in the defining property come from the left-quotient sets of X, Y, and Z, respectively. The subgroup version is less general and is used for illustrative purposes.

While $\omega_s \leq \omega$ is obvious, it is not at all clear that an upper bound on ω_s implies anything about ω. Our main technical contribution is the following theorem:

Theorem 10 ([5]). *The following inequality holds: $\omega \leq (3\omega_s - 2)/2$.*

In particular, if $\omega_s \leq 2+\epsilon$, then $\omega \leq 2 + (3/2)\epsilon$, and thus $\omega_s = 2$ implies $\omega = 2$.

Armed with this theorem we can begin exploring general semi-simple algebras. A promising family of algebras, with the requisite "nice basis" and underlying combinatorial/algebraic object, are *adjacency algebras* of *coherent configurations*. We do not have room for definitions here, but one can think of coherent configurations as a common generalization of groups and group-actions. Every coherent configuration C has an easily discernible *rank*, and an associated (semi-simple) *adjacency algebra* $\mathcal{C}[C]$ analogous to the group algebra. If the coherent configuration is commutative, the adjacency algebra is as well, which implies that the rank of the $\mathcal{C}[C]$-multiplication tensor is simply the rank of the coherent configuration.

Coherent configurations arising from group actions are called *Schurian*, and one can state a sufficient condition for these to realize matrix multiplication, similar to the triple product property. If G acts on set X, and A, B, C are subsets of X for which

$$fa \in A \wedge gb \in B \wedge hc \in C \Rightarrow fa = a \wedge gb = b \wedge hc = c$$

for all $a \in A, b \in B, c \in C$ and $fgh = 1$, then the Schurian coherent configuration associated with this group action realizes $\langle |A|, |B|, |C| \rangle$.

Finally, we show in [5] that non-trivial bounds on ω_s *can* be achieved via *commutative* Schurian coherent configurations (in contrast to the group setting). And, if either Conjecture 3 or Conjecture 4 are true, then commutative coherent configurations suffice to prove $\omega_s = 2$ (which then implies $\omega = 2$).

The main message is this: embedding $n \times n$ matrix multiplication into a commutative coherent configuration of rank $n^{2+o(1)}$ is a viable route to proving $\omega = 2$.

References

[1] N. Alon, A. Shpilka, and C. Umans. On sunflowers and matrix multiplication. Proceedings of the 27th IEEE Conference on Computational Complexity, 26–29 June 2012, Porto, Portugal, IEEE Computer Society, pp. 214–223.

[2] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[3] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans. Group-theoretic algorithms for matrix multiplication. Proceedings of the 46th Annual Symposium on Foundations of Computer Science, 23–25 October 2005, Pittsburgh, PA, IEEE Computer Society, pp. 379–388.

[4] H. Cohn and C. Umans. A group-theoretic approach to fast matrix multiplication. Proceedings of the 44th Annual Symposium on Foundations of Computer Science, 11–14 October 2003, Cambridge, MA, IEEE Computer Society, pp. 438–449.

[5] H. Cohn and C. Umans. Fast matrix multiplication using coherent configurations. To appear in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA), 2013.

[6] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computation*, 9:251–280, 1990.

Lower bounds against ACC circuits

Ryan Williams

We gave an overview of the recent proof [4] that nondeterministic exponential time (NEXP) does not have constant-depth circuits of polynomial-size comprised of AND, OR, and modulo-m gates of unbounded fan-in, for some constant m. (This circuit class is often called ACC; see [1].) Several new simplications and extensions have been made since the original proof was announced. Probably the three most significant ones are:

(1) a simpler and more practical algorithm for solving ACC circuit satisfiability, using a divide-and-conquer approach (see [3]),

(2) a simple way to construct polynomial-size ACC circuits from arbitrary $O(\log n)$-depth circuits in subexponential time, under the assumption that LOGTIME-uniform NC^1 is contained in polynomial-size ACC (see [2]), and

(3) a new argument, building on the prior one, that can be used to extend the ACC lower bounds down to the class $\mathsf{NEXP} \cap \mathsf{coNEXP}$ (forthcoming).

References

[1] R. Beigel and J. Tarui. *On ACC*, Computational Complexity **4** (1994), 350–366.

[2] R. Santhanam and R. Williams, *Uniform circuits, lower bounds, and QBF algorithms*, Electronic Colloquium on Computational Complexity (ECCC) **19** (2012), report 59.

[3] R. Williams. *Guest column: a casual tour around a circuit complexity bound*, SIGACT News **42(3)** (2011), 54–76.

[4] R. Williams, *Non-uniform ACC circuit lower bounds*, Proceedings of IEEE Conference on Computational Complexity (2011), 115–125.

Interactive Proofs for Delegating Computation

Guy N. Rothblum

(joint work with Shafi Goldwasser, Yael T. Kalai, Salil Vadhan, Avi Wigderson)

The power of efficiently verifiable proof systems is a central question in the study of computation. We study efficiently verifiably *interactive proof* systems, as introduced by Goldwasser, Micali and Rackoff [GMR89]. We focus on interactive proof systems where the verifier is super-efficient, and *the honest prover is also efficient*: For a given language, we seek an interactive proof where computational power (time, depth, or space) required for verifying the proof is smaller than power needed to compute the language. We call this *super-efficient verification.* In addition, the power needed for computing the proof (e.g. the running time), i.e. for running the honest prover's algorithm, is polynomial in the power needed to compute the language (e.g. the running time of the best algorithm known). We call this the *efficient proof* property.

Beyond its importance as a foundational question, the study of interactive proofs with an efficient proof and super-efficient verification is motivated by applications to delegating computation. In that setting, several computational devices of differing computational abilities interact with each other over a network. Some of these devices are computationally weak due to various resource constraints. As a consequence there are tasks, which potentially could enlarge a device's range of application, that are beyond its reach. A natural solution is to *delegate* computations that are too expensive for one device, to other devices which are more powerful or numerous and connected to the same network. The fundamental problem that arises is: *how can a delegator verify that the delegatees performed the computation correctly, without running the computation itself?* Interactive proofs systems with an efficient proof and super-efficient verification provide a solution to this problem.

In this abstract, we highlight two main results in the study of interactive proofs with an efficient proof and super-efficient verification. Both of these results focus on super-efficient verification in terms of the verifier's running time (together with the efficient proof property). We note that in further works on this question, Goldwasser *et al.* [GGHKR07] considered super-efficient verification in terms of the verifier's depth, and Goldwasser, Kalai and Rothblum [GKR08] considered super-efficient verification in terms of the verifier's space.

Efficient Interactive Proofs for Bounded-Depth Computations. Our main result gives interactive proofs for general uniform computations:

Theorem 1 (Goldwasser, Kalai and Rothblum [GKR08].)**.** *Take* $S = S(n)$*,* $D = D(n)$ *Let* L *be a language that can be computed by a family of* $O(\log S)$*-space uniform*[5] *boolean circuits of size* S *and depth* D*.* L *has an interactive proof where:*

(1) *The prover runs in time* $poly(S)$*, while the verifier runs in time* $n \cdot poly(D, \log S)$ *and space* $O(\log S)$*.*
(2) *The protocol has perfect completeness and soundness* $1/2$.[6]
(3) *The protocol is public-coin, with communication complexity* $D \cdot \mathrm{polylog}(S)$*.*

Remark 2 (Interpreting Theorem 1)**.** *One particular setting of parameters for Theorem 1 is languages that are in (log-space uniform)* $\mathcal{NC}$*: languages computable by (uniform) circuit of size* $poly(n)$ *and depth* $\mathrm{polylog}(n)$*. For this rich class of languages, Theorem 1 gives an interactive proof system where the prover runs in time* $poly(n)$*, the verifier runs in time* $\tilde{O}(n)$ *and space* $O(\log n)$*, and the communication complexity is* $\mathrm{polylog}(n)$*.*

Comparison to Prior Work on Interactive Proofs. We note that Theorem 1 improves previous work on interactive proofs [LFKN92, Sha92, FL93], in terms of the honest prover's running time. In particular, interactive proof systems proposed in these prior works did not have the *efficient proof* property (for any non-trivial family of languages).

Sublinear-Time Verification for Bounded-Depth Computations. We also consider interactive proofs with efficient proofs and *sublinear time* verifiers. These proof systems can be used by a sublinear-time client for delegating computation:

[5] A circuit family is $s(n)$-space uniform if there exists a Turing Machine that on input 1^n runs in space $s(n)$ and outputs the circuit for inputs of length n.

[6] Throughout this work we work with constant soundness for interactive proof systems. This is easily amplified via parallel or sequential repetition.

the client, who only has (reliable) *query* access to a potentially huge input, can interact with a powerful but unreliable server who has full access to the input. The client can delegate its computations to the server, and receive a proof of approximate correctness for the results.

As in the study of sublinear time algorithms, randomness is essential. Following the literature on property testing [GGR98], we seek proof systems where with high probability the verifier accepts every input in the language, and rejects every input that is ε-*far* from the language, where $\varepsilon = \varepsilon(n)$ is the fractional Hamming distance (and can be a function of the input length n). The verifier's query complexity (and computation complexity), as well as the communication, should all be sublinear. We call such a proof system an *Interactive Proof of ε-Proximity.* Building on Theorem 1, we show interactive proofs of proximity for general uniform computations:

Theorem 3 (Rothblum, Vadhan and Wigderson [RVW12].)**.** *Take $\varepsilon = \varepsilon(n) \in (0,1)$, $S = S(n)$, $D = D(n)$ Let L be a language that can be computed by a family of $O(\log S)$-space uniform*[7] *boolean circuits of size S and depth D. L has an interactive proof of ε-proximity where:*

(1) *The prover runs in time poly(S). The verifier runs in time $(\varepsilon \cdot n + 1/\varepsilon)^{1+o(1)} \cdot$ poly$(D, \log S)$.*
(2) *The protocol has perfect completeness and soundness $1/2$.*
(3) *The protocol is public-coin, with communication complexity $(\varepsilon \cdot n \cdot$ poly$(D) \cdot (1/\varepsilon)^{o(1)})$ and verifier query complexity $(1/\varepsilon)^{1+o(1)}$.*

Remark 4 (Interpreting Theorem 3)**.** *In interpreting this result, we can again consider the class of $\mathcal{NC}$ languages (see Remark 2). For such languages, Theorem 3 gives a tradeoff between the query complexity q and communication complexity c, where $q \times c = n^{1+o(1)}$. The number of rounds is poly-logarithmic, and the honest prover runs in polynomial time. For example, for any $\varepsilon \geq n^{-1/2}$, the queries, communication, and verifier runtime can all be be $n^{1/2+o(1)}$.*

Comparison to Prior Work on Sublinear-Time Computation and Verification. A rich body of work within the literature on sublinear time algorithms focuses on *property testing* [RS96, GGR98]. There, a randomized tester has query access to the input, and needs to distinguish whether the input is in a language or far from the language. We extend this model by also providing interaction with a more powerful prover, who can read the input in its entirety, but might cheat.

Another beautiful line of research, starting with the work of Babai *et al.* on Holographic Proofs [BFLS91], has focused on "PCP-like" proof systems with sublinear time verifiers. Work on PCP spot checkers [EKR04], PCPs of Proximity [BGHSV06] and Assignment Testers [DR06], extended the property testing model by giving the verifier query access to a *fixed* proof string. While that proof string

[7] A circuit family is $s(n)$-space uniform if there exists a Turing Machine that on input 1^n runs in space $s(n)$ and outputs the circuit for inputs of length n. A circuit family is $\mathcal{L}$-uniform if it is log-space uniform.

may be wrong, it is nonetheless fixed and does not change with verifier's queries to it. In our model (and in the delegating computation motivation) the prover can *adaptively* change its strategy and answers, as a function of the verifier's messages.

References

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. *STOC*, pages 21–31, 1991.

[BGHSV06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the pcp theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.

[EKR04] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. *STOC*, pages 113–122, 2008.

[GGHKR07] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. Verifying and decoding in constant depth. *STOC*, pages 440–449, 2007.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[FL93] Lance Fortnow and Carsten Lund. Interactive proof systems and alternating time-space complexity. *TCS*, 113(1):55–73, 1993.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RVW12] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson . Interactive Proofs of Proximity: Delegating Computation in Sublinear Time. *Manuscript*, 2012.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

Towards provable bounds for machine learning—three vignettes

Sanjeev Arora

(joint work with Rong Ge, Ravi Kannan, Ankur Moitra, Sushant Sachdeva)

Many tasks in machine learning (especially unsupervised learning) are provably intractable: NP-hard or worse. Nevertheless, researchers have developed heuristic algorithms to try to solve these tasks in practice. In most cases, these algorithms are heuristics with no provable guarantees on their running time or on the quality of solutions they return. Can we change this state of affairs?

After surveying machine learning for a theoretical CS audience, this talk suggests that the answer is yes, and describe three of our recent works as illustration. (a) A new algorithm for learning topic models. (It applies to Linear Dirichlet Allocations of Blei et al. and also to more general topic models. It provably works under some reasonable assumptions and in practice is up to 50 times faster than existing software like Mallet. It relies upon a new procedure for nonnegative matrix factorization.) (b) What classifiers are worth learning? (Can theory illuminate the contentious question of what binary classifier to learn: SVM, Decision tree, etc.?) (c) Provable ICA with unknown gaussian noise. (An algorithm to provably learn a ”manifold” with small number of parameters but exponentially many ”interesting regions.”)

References

[1] S. Arora, R. Ge, A. Moitra. Learning Topic Models – Going Beyond SVD. IEEE 53rd Annual Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick NJ, USA, October 20-23. pp. 1-10.

[2] S. Arora, R. Ge, R. Kannan, A. Moitra. Computing a Nonnegative Matrix Factorization – Provably. Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA. pp 145-162.

[3] S. Arora, R. Ge, S. Sachdeva, A. Moitra. Provable ICA with Unknown Gaussian Noise, and Implications for Gaussian Mixtures and Autoencoders. Advances in Neural Information Processing Systems (NIPS) 2012.

Reporter: Stefan Mengel