Research Statement

Klim Efremenko

March 9, 2017

1 Overview

My main focus of research is in Error Correcting Codes and I have a broad interest in theoretical computer science and mathematics. Error Correcting Codes is a classical research area, starting with the works of Shannon, Huffman in the 50s, and it has been remarkably successful at addressing some of its original challenges.

My work is focused on two emerging areas, with both theoretical and practical importance: Error Correction for Interactive Communication, and Locally Decodable Codes (LDCs). Both of these areas emerged out of theoretical research, and contain deep mathematical challenges. On the other hand, they are becoming increasingly important in a number of applications, which in turn lead to more intriguing theoretical and algorithmic problems. These topics are also highly related to the current trend of cloud computing and storage.

My main contributions are: In the area of Locally Decodable Codes I have settled down a long standing open problem namely construction of sub-exponential LDCs with constant query. I also established a tight connection between Locally Decodable Codes and the representation theory. In the area of Interactive Communication my main contributions are: a complete characterization of the error rates which can be handled via coding for interactive communication and lower and upper bounds on the rate of interactive communication in the networks.

Coding for Interactive Communication

Classic error correcting codes are designed to encode messages sent from one party to another. They are optimised to correct a large number of errors, while still having efficient encoding and decoding algorithms. Most modern communication is interactive, where two or more parties are communicating and responding to each other’s messages. However, the coding schemes being used are still those which were optimised decades ago for single messages. This creates the opportunity to design better codes for interactive communication. To do so we analyse the capacity of interactive channels and the limitations of coding in interactive settings.

Together with my co-authors in [BE14] I solved one of the important open problems in this field: a complete characterisation of the error rates which can be handled via coding
for interactive communication. The result is very surprising since the region we get is not smooth and without the exact calculation it is hard to guess it.

Another question that I have been studying recently is interactive communication in networks. In [BEGH16] we have shown first lower bounds on the rate of communication in networks with noise. In this work we have showed that how sensitive network to noise highly depends on its topology. For example we showed in [ABE+16] that in a network where everyone connected to everyone slowdown due to random noise is only constant while if a network has a star topology then there would be a slowdown of order log n where n is number of parties participating in the communication. In the paper [AES16] we show how to use tools from additive combinatorics to design new protocols for an equality problem on network.

One of the problems of coding for interactive communication is that in while it preserves communication complexity in general it may blow up the number of rounds arbitrarily. In the paper [EHK16] we show how to construct coding scheme for interactive communication that increases both communication and round complexity of the protocol only by a constant factor.

The field of coding for interactive communication is very young and strongly developing and there are still many challenges remaining, which I am currently investigating.

Locally Decodable Codes  The straightforward approach in coding theory to decoding a message is to first read the entire received word (which is a corrupted codeword), process it as efficiently as possible, and output the best estimate for the correct codeword. An obvious limitation of this paradigm is the requirement to read and process the entire received word. Locally decodable codes comes to solve this issue. They allow one to extract useful information about the correct codeword (such as certain bits in it) while reading and processing only a tiny fraction of the received word. While this research was initially motivated by theoretical applications, such as the celebrated PCP theorem, it has found some surprising real life applications, such as privacy in databases (PIR schemes) and recently LDCs where implemented for recovery from node failure in distributed storage in Microsoft Azure. Still, we have very poor understanding of the potential of locally decodable codes.

I gave the first unconditional construction of a sub-exponential locally decodable code in the regime of constant query complexity, which some had conjectured was impossible. I also gave algorithms to decode and list decode these codes and investigated better approaches for the design of such codes via mathematical problems in representation theory.

Recently I have started to work on problems related to LDCs and distributed data storage. Specially we have showed a coding that allows fast disc recovery in the case where a small number of discs were damaged.

Arithmetic Circuits  Arithmetic circuits is a model for computing polynomials. Informally we start from variables and we are allowed to multiply divide and add previously computed polynomials. Model of arithmetic circuits captures many important computations such as Fast Fourier Transform, Error Correcting Codes, most linear algebra computations
and many others. In the papers [ELSW16, ELSW15, EGdOW] I am using tools from com-
mutative algebra and algebraic geometry to study arithmetic circuits.

2 Interactive Communication

The area of Error Correcting Codes (ECC) deals with the question of how to transmit a
message over a noisy channel. The standard setting is when Alice wants to send a message
to Bob over a channel with noise. There are two main models of noise. The Shannon
model, is where the noise is random, where each bit is flipped with some probability $p$. The
Hamming model, is where the noise is adversarial, in which any $pn$ bits may be corrupted
(here $n$ is the length of the message). We call $p$ here an error rate of the code. An error
correcting code is a map $C : \{0, 1\}^k \rightarrow \{0, 1\}^n$ which maps a message $m$ to a codeword $C(m)$
such that the noisy version of $C(m)$ allows recovery of $m$. The Hamming model of noise
is more restrictive and this makes it harder to construct ECC for it, but it also makes it
much more useful in applications of theoretical computer science. The ratio $k/n$ is called
the transmission rate (or just rate) of the code. A family of codes is called good if it can
achieve constant transmission and error rate. For example one can show that over a random
channel there exist a good family of codes as long as the error rate $p$ has $p < 1/2$ and over
an adversarial channel good family of codes exists only if error rate $p < 1/4$. One way
to close the gap is to use list decodable codes. In this scenario instead of decoding to one
answer Bob outputs a small (ideally constant) list of possible answers. In this situation we
can handle any $p < 1/2$ fraction of noise in the adversarial model (for more details see the
book [Gur01]).

Optimal Error Rates for Unique and List Decoding

A very natural generalization of error correcting codes is to understand the situation when, instead of sending just one
message, Alice and Bob want to perform an interactive communication. It is not at all
obvious that good error-correction is possible against adversarial substitution errors of any
constant rate. Note that any attempt to applying standard error-correcting codes round-
by-round are bound to fail, since all the adversary has to do to derail the execution of the
protocol is to completely corrupt a single round. Therefore, a sub-constant error rate of $1/r$
suffices to foil an $r$-round protocol protected with a round-by-round code.

In a seminal work, Schulman [Sch96] showed that there exist good error-correcting codes
for interactive communication against adversarial error rate $p$ with $p < 1/240$. Interest
in interactive error correction has been renewed recently, with Braverman and Rao [BR11]
showing that the error rate that can be tolerated can be improved to $p < 1/4$. Unfortu-
nately both constructions of [Sch96] and [BR11] are not computationally efficient, but series
of recent works made significant progress toward making interactive error correction compu-
tationally efficient [GMS11, Bra12, BK12, FGOS12, BN13, GHS14, GH14]. Essentially
in last two works by Ghaffari, Haeupler, and Sudan [GHS14, GH14] it was shown that if
one can perform (non-efficiently) list decoding for interactive communication at an optimal
error rate and constant transmission rate then there exist efficient unique and list coding for
interactive communication at an optimal error rate.

In joint work with Braverman [BE14] we investigate the model for interactive communication with adversarial noise. We first develop a notion of interactive list decoding \(^1\), which is the list analogue of interactive error correction. That is after the execution of a protocol, each party will output a constant-size list of possible original conversations. If the fraction of errors \(\leq p\), each list will contain the intended conversation. We show that constant (transmission) rate interactive list decodable coding is possible for all error rates \(p < 1/2\) as well we show that for \(p \geq 1/2\) rate goes exponentially fast to zero. Specifically we show the following theorem:

**Theorem 2.1** ([BE14]). For each \(\varepsilon > 0\) and for every protocol \(\pi\), there exists another protocol \(\pi'\), with communication complexity only constant times larger, that is resilient to \(1/2 - \varepsilon\) adversarial noise. The protocol \(\pi'(x,y)\) outputs a list of size \(1/\varepsilon^3\) of transcripts such that \(\pi(x,y)\) is in the list.

Besides its application to construct computationally efficient schemes interactive list decoding turns out to be the right tool for giving tight bounds on the error rates we can tolerate in the unique decoding setting. In the interactive setting, it is natural to consider pairs \((\alpha, \beta)\) of error rates, with \(\alpha\) representing the fraction of Alice’s communication that may be corrupted and \(\beta\) representing the fraction of Bob’s communication that may be corrupted. Using our list decoding results, we are able to give a precise characterization of the region \(R_U\) of pairs \((\alpha, \beta)\) of error rates for which constant-rate unique decoding is possible. The region \(R_U\) turns out to be unusual in its shape. In particular, it is bounded by a piecewise-differentiable curve with infinitely many pieces.

Let me mention that the question of extending above results is widely open.

**Open Problem 1.** The question of understanding maximal error rate is open over a binary alphabet. The maximal noise it is possible to tolerate over a binary alphabet is known to be between \(1/8\) and \(1/6\).

**Interactive Communication in Networks**  Next interesting question we want to ask is what happens if we have more than two parties connected by a network who are trying to perform an interactive task. This question was studied by Rajagopalan and Schulman [RS94] where they have shown that one can perform interactive communication in a network in presence of random noise by loosing only factor \(O(\log d)\) in the time (here \(d\) is the maximal degree of node of the graph of connections of the network). For example if \(n\) people are connected by a complete graph noise will slow down communication by a factor of \(O(\log n)\).

It was believed that the factor \(\log n\) is necessary. However no lower bounds for interactive communication over networks where known before. In work [ABE+16] we show that the rate of interactive protocol depends not only on its degree but also on its spectral properties. For example over a complete graph (more generally over a graph with constant mixing time)

\(^1\)In independent work [GHS14] has also developed notion of list decoding but they where not able to achieve constant rate.
one can perform interactive communication with noise with slow down only by a constant factor, disproving the conjecture that the factor \( \log n \) is necessary for a complete graph. We showed even more that for graphs with small mixing time there exist an efficient interactive protocols. On other hand in work [BEGH16] we show that if \( n \) people are connected by a star network, interactive communication will be slower by a factor \( \Theta(\frac{\log n}{\log \log n}) \), which is the first lower bound for multi-party interactive communication.

3 Locally Decodable Codes

A code \( C \) is said to be Locally Decodable Code (LDC) with \( q \) queries if it is possible to recover any symbol \( x_i \) of a message \( x \) by making at most \( q \) queries to \( C(x) \), such that even if a constant fraction of \( C(x) \) is corrupted, the decoding algorithm returns the correct answer with high probability.

Despite the importance of LDCs for practical questions, my interest in their study comes from its applications to complexity theory and cryptography. Many important results in these fields rely on LDCs. LDCs are closely related to such subjects as worst case - average case reductions, pseudo-random generators, hardness amplification, and private information retrieval schemes (see for example [PS94, Lip90, CKGS98, STV01, Tre03, Tre04, Gas04]). Locally Decodable Codes also found applications in data structures and fault tolerant computation, see for example [CGdW10, dW09, Rom06].

Locally Decodable Codes implicitly appeared in the Probabilistically Checkable Proofs literature already in early 1990s most notably in [BFLS91, PS94, Sud92]. However the first formal definition of LDCs was given by Katz and Trevisan [KT00] in 2000. Since then LDCs have become a widely used notion. The first constructions of LDCs [BIK05, KT00] were based on polynomial interpolation techniques. Later on more complicated recursive constructions were discovered [BIKR02, WY07]. But all these constructions have exponential length. Tight lower bound of \( 2^{O(n)} \) codes were given in [KdW04, GKST06] for two query LDCs. For many years it was conjectured (see [Gas04, Gol05]) that LDCs should have an exponential dependence on \( n \) for any constant number of queries, until Yekhanin’s breakthrough [Yek08]. Yekhanin obtained 3-query LDCs with sub-exponential length. Unfortunately, Yekhanin’s construction is based on an unproven but a highly believable conjecture in number theory and this construction is lacks an intuitive explanation.

Current Results In [Efr12a] I gave the first unconditional construction of sub-exponential LDCs what is completely refutes the conjecture that constant query LDCs must have exponential length.

Theorem 3.1 ([Efr12a]). For every \( r \) and for every \( k \) there exists a \( 2^r \) query LDC \( C : \mathbb{F}^k \rightarrow \mathbb{F}^n \), where \( n = \exp(\exp(O(\sqrt{\log n(\log \log n)^{r-1}}))) \).

This construction is based on the combinatorial object called matching vectors. Matching vectors is an interesting object from extremal combinatorics which has very surprising prop-
erties when working over composite numbers rather than over primes. In the paper [Efr12a] in fact we show the following relation between matching vectors and LDCs.

**Theorem 3.2 ([Efr12a]).** For every $S$-Matching Vectors $\{u_i\}_{i=1}^k \subset \mathbb{Z}_m^k$ there exists an $|S| + 1$ query LDC $C : \mathbb{F}^k \rightarrow \mathbb{F}^m$.

This gives first unconditional construction of LDCs of sub-exponential length. This construction explains Yekhanin’s construction, eliminates the dependence on number theoretic conjectures and improves the parameters of the code. I would like to mention a recent very surprising result, where based Matching Vector construction of LDCs Dvir and Gopi [DG14] gave a Private Information Retrieval scheme with two servers and sub-polynomial communication. Before this work was was highly believed that no such scheme exist.

The history of Matching Vectors is similar to the history of LDCs. It was conjectured for many years that there must be a polynomial upper bound on the size of MV, until Grolmusz’s [Gro00] breakthrough. Today there is not even a conjecture of what are the best possible parameters for LDCs and MVs. Therefore, although the construction in [Efr12a] is a simple it still does not lead us to optimal LDCs. This leads us to seek a new approach to understanding LDCs. In [Efr12b] I initiated a systematic study of LDCs from the point of view of the representation theory of finite groups. In [Efr12b] I showed the tight connection between LDCs and irreducible representations in the following theorem

**Theorem 3.3 ([Efr12b]).** Let $G$ be a finite group and let $(\rho, \mathbb{F}^k)$ be an irreducible representation of $G$ and there exist $q$ elements $g_1, g_2, \ldots, g_q$ in $G$ such that some linear combination of the matrices $\rho(g_i)$ is a rank one matrix. Then there exists a $q$ query LDC $C : \mathbb{F}^k \rightarrow \mathbb{F}^G$.

This approach unifies known constructions of LDCs and reveals what I believe is the real algebraic nature behind LDCs. Although the question of how to construct such representations is a natural one, it was never considered before. I believe that this study of LDCs will bring many natural and interesting questions to representation theory and that both fields will benefit from this study.

Below are two problems which I found interesting and I am working on them now.

**Open Problem 2.** Construct natural representations that will lead to a sub-exponential LDCs.

**Open Problem 3.** Extend the above framework to the world of the modular representations.

### 4 Arithmetic Circuits

I also have a deep interest in study arithmetic analogue of the problem $P$ vs $NP$ namely the problem $VP$ vs $VNP$. This algebraic model of computation attracted a substantial amount of research in the last five decades, partially due to its simplicity and elegance. Being a more structured model than Boolean circuits, one could hope that the fundamental problems of theoretical computer science, such as separating P from NP, will be easier to
solve for arithmetic circuits. However, in spite of the apparent simplicity and the vast amount of mathematical tools available, no major breakthrough has yet arisen. In fact, all the fundamental questions are still open for this model as well. Nevertheless, there has been a progress in the area and beautiful results have been found, some in the last few years.

It turns out that many other problems in arithmetic complexity have natural analogues in algebraic geometry. For example, the study of depth 3 circuits (a poorly understand model of algebraic computation) is equivalent to the study of secant varieties of the Chow variety (a classic variety from algebraic geometry). Until recently, the foremost technique for proving lower bounds was the method of partial derivatives, as introduced by [NW97]. Recently a generalization of this approach was introduced in [Kay12] and the power of this method was first fully demonstrated in [GKKS13]. Method of shifted partial derivatives already has found large number of applications in arithmetic circuits see [KSS14, KLSS14b, KLSS14a, FLMS14, KS14]. In the paper [ELSW16] we show why it is unlikely that method of shifted partial derivatives will show that $\text{VP} \neq \text{VNP}$.

In the papers [ELSW15] together with J.M. Landsberg, Hal Schenck, and Jerzy Weyman I study two promising approaches to extend shifted partial derivatives. The first approach comes from representations of reductive groups called Young Flattening. It was first introduced in [LO13] where this approach has already shown its power by proving the best lower bounds for matrix multiplication. The method of shifted partial derivatives is a special case of the study of Young Flattenings. Given the power of shifted partial derivatives, and that Young flattenings generalized this method, I believe that now it is the right time to explore the power of Young Flattenings applied to different circuit classes. The second way to generalize shifted partial derivatives is to study the minimal free resolution of the ideals generated by partial derivatives (see books [Eis06, Wey03]). If one has the free resolution of the ideal generated by partial derivatives of some polynomial then one can extract from it the dimensions of shifted partial derivatives. Using tools from representation theory we can compute the free resolution of the permanent. One of the consequences of this result is that we can compute the dimensions of the shifted partial derivatives of the permanent which was not known before (see [GKKS13]). This result has independent interest for algebraic geometry.

In the paper [EGdOW] together with Ankit Garg, Rafael de Oliveira and Avi Wigderson we study what lower bounds can give us rank techniques. Rank technique is a very common way of proving lower bounds: where one associate for every polynomial a matrix in a way that ”easy” polynomials will have low rank and hard polynomials will have high rank. We show that such methods can not improve lower bounds for tensor rank and for waring rank by much. As well we show the way one can try to find a measure for lower bounds for depth-3 circuits.

5 Other Results

I am broadly interested in algorithms and in applications of mathematical tools to study them.
**Pattern Matching**  The most classical problem in pattern matching is: given a text of length $n$ and a pattern of length $m$ find all occurrences of the pattern in the text. An almost optimal solution for this problem was found already in the 1970s. This problem has two natural extensions: first one is where the text and pattern contain special symbols “don’t care” which match all other symbols and the second one is to find all occurrences of a pattern with at most $k$ mismatches. For each one of these problems there exists a good solution, but unfortunately there was no efficient solution which can handle both of these extensions. In [CEPR10] we show a randomized algorithm that runs in time $\tilde{O}(nk)$ and find all occurrences of the pattern in the text with at most $k$ mismatches even when both pattern and text may contain “don’t care”. Our approach is based on the sampling technique. We show how to sample a random mismatch at every location in time $\tilde{O}(n)$. Running the sampling algorithm $\tilde{O}(k)$ times gives us a randomized algorithm for $k$ mismatch problem with “don’t cares”. Later on in [CEPR09] we found the large similarity between the sampling technique and efficient encoding and decoding algorithms for Reed-Solomon codes. Using the tools developed for efficient encoding and decoding of Reed-Solomon codes we give a deterministic algorithm this problem that runs almost the same time as randomized algorithm.

The importance of sampling technique comes not only from its applications to the problem of $k$ mismatches, but also since it could be used also in other variants of pattern matching. For example in [EP08] we use this technique for the following problem: assume that all symbols from pattern and text comes from some metric space. The question is how fast can we approximate sum of the distances between symbols of pattern and text for all possible offsets of the pattern. In order to solve this problem we use sampling technique and tools from computational geometry.

**Random Walks**  A random walk on a graph is a process that explores the graph in a random way: at each step the walk is at a vertex of the graph, and at each step it moves to a uniformly selected neighbor of this vertex. Random walks are extremely useful in computer science and in many other fields. Main parameters of the random walk are the time it takes to visit all nodes in the graph (cover time), the time it takes to reach some specific node in the graph (hitting time). A very natural problem is to analyze the behavior of $k$ independent walks in comparison with the behavior of a single walk. Inspired by [AAK+11] in [ER09] we initiate a systematic study of multiple random walks. We show that behavior of random walks highly depends on the starting points. We give lower and upper bounds both on the cover time and on the hitting time of multiple random walks over three alternatives for the starting vertices of the random walks: the worst starting vertices (those who maximize the hitting/cover time), the best starting vertices, and starting vertices selected from the stationary distribution. As a rather surprising corollary of our theorems, we obtain a new bound which relates the cover time and the mixing time of a single random walk.
References


