Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 123 (2010)

Limitation on the rate of families of locally testable codes

Eli Ben-Sasson
Computer Science Department
Technion — Israel Institute of Technology
Haifa, 32000, Israel.
eli@cs.technion.ac.il

August 11, 2010

Abstract

This paper describes recent results which revolve around the question of the rate attainable by families of error correcting codes that are locally testable. Emphasis is placed on motivating the problem of proving upper bounds on the rate of these codes and a number of interesting open questions for future research are suggested.

1 Introduction

A locally testable code (LTC) is an error correcting code for which membership in the code can be ascertained, to a high degree of confidence, by a random process that queries a negligible fraction of a purported codeword. Locally testable codes, first studied by Blum, Luby, and Rubinfeld [1990], lie at the core of all known constructions of probabilistically checkable proofs (PCPs), from [Arora and Safra, 1998, Arora et al., 1998] to [Dinur, 2007], their discovery has inspired the study of property testing [Goldreich, Goldwasser, and Ron, 1998], and the construction of such codes has been of great interest to theoretical computer science in the recent past. Several surveys describe the concepts around which these codes revolve [Goldreich, 2005, Trevisan, 2004], and a number of distinct ways to obtain such codes are known by now (see Section 1.2). The purpose of this brief survey, which assumes familiarity with the basic notion of an LTC, is to explain what is known about the *limitations* of constructions of such codes, or, in plain words, what kinds of LTCs are mathematically impossible to obtain.

When studying locally testable codes we are interested in both the classically studied parameters of error correcting codes, such as *rate* and *relative distance*, as well as in the local-testability parameters of the code, the *query complexity* or number of entries read by the testing process, and the *completeness* and *soundness* which measure the probability of correctness of this process (these concepts are defined in the next subsection). We intend to study the interplay between these two kinds of code-related parameters so let us informally explain what kind of trade-offs we expect to see. Better local-testability parameters, like smaller query complexity and larger completeness and soundness parameters should be expected to negatively affect the classical coding parameters, decreasing the rate and/or relative distance of the code. We can show that this intuition does indeed hold for certain families of codes, as surveyed later on. But for all the effort that has gone into the study of LTCs, the fundamental question that motivates our study (Question 1.6), regarding the

ISSN 1433-8092

existence of an asymptotically good family of LTCs, remains wide open. Before we continue we pause to recall the definition of a locally testable code and the reader familiar with this definition and the associated notation is encouraged to skip the following subsection.

1.1 Defining locally testable codes

We assume familiarity with the basic definitions of error correcting codes, which can be found, e.g., in MacWilliams and Sloane [1978]. A *code* $\mathcal{C}$ over alphabet Σ of *blocklength* n, *message-length* k and *minimal distance* d will be called an $(n, k, d)_\Sigma$-code. It is a subset of Σ^n of size at least $|\Sigma|^k$ which satisfies the condition that for any pair of distinct codewords $w, w' \in \mathcal{C}$ their *Hamming distance*, defined as the number of entries on which w and w' disagree, is at least d. We shall reserve the letter w to denote codewords and r to denote "received" words, words which are not known to belong to $\mathcal{C}$. The i^{th} entry of r will be denoted by r_i.

Two fundamental parameters of a code are its *rate* $\rho(\mathcal{C}) = k/n$ which measures the ratio of message to codeword length and the *relative distance* $\delta(\mathcal{C}) = d/n$ which dictates the noise-resilience of the code. We shall be interested in *families of codes* $\{\mathcal{C}_n \subset \Sigma^n \mid n \in \mathbb{Z}\}$. A family of codes is said to be *asymptotically good* if all members of it have positive rate and relative distance, i.e., there exist constants $\rho, \delta > 0$ such that each $\mathcal{C}_n$ satisfies $\rho(\mathcal{C}_n) \geq \rho$ and $\delta(\mathcal{C}_n) \geq \delta$. Given $\mathcal{C}$ and $r \in \Sigma^n$ let $\delta_{\mathcal{C}}(r)$ denote the *relative (Hamming) distance* between r and $\mathcal{C}$, defined as the minimal fraction of entries of r that need to be changed in order to obtain a word in $\mathcal{C}$. When $\delta_{\mathcal{C}}(r) \geq \epsilon$ we say r is *δ-far* from $\mathcal{C}$ and otherwise say r we say *ϵ-close* to it.

When Σ is the q-element finite field $\mathbb{F}_q$ (when the size of $\mathbb{F}$ is known or insignificant we use $\mathbb{F}$ to denote it) and $\mathcal{C}$ is a *linear code*, i.e., a k-dimensional subspace of $\mathbb{F}^n$, we shall say $\mathcal{C}$ is an $[n, k, d]_{\mathbb{F}}$-code. In this case the distance of the code is equal to the minimal weight of a nonzero codeword, where the *weight* of a word $r \in \mathbb{F}^n$ is the number of nonzero entries in r.

A *locally testable code* is an error correcting code — we expect it to have large relative distance — which comes with a randomized algorithm, called a *tester*, that samples a small number of entries of a received word $r \in \Sigma^n$ and is capable of distinguishing with nontrivial probability between the "good" case that r is an uncorrupted codeword, i.e., that r belongs to $\mathcal{C}$ (so $\delta_{\mathcal{C}}(r) = 0$) and the "bad" case that r is ϵ-far from $\mathcal{C}$. Since the definition of an LTC is tied to that of a tester we give both of them together.

Definition 1.1 (Tester and locally testable code)**.** Let $\mathcal{C}$ be an $(n, k, d)_\Sigma$-code. A *(q, ϵ, s, c)-tester* for $\mathcal{C}$ is a randomized algorithm T with oracle access to a purported codeword $r \in \Sigma^n$ which operates as follows. The tester T uses randomness to sample at most q entries of r and outputs a verdict which is either accept or reject. Denote by $T^r[R]$ the output of T on oracle r and random coins R. We say that T is a *q-query tester*, or, simply, a *q-tester*.

The code $\mathcal{C}$ is said to be *(q, ϵ, s, c)-locally testable* if it has a q-tester that satisfies the following *completeness* and *soundness* requirements. It the tester satisfies the (stronger) requirement of *strong soundness* we say $\mathcal{C}$ is a *(q, s, c)-strong locally testable* code.

Completeness If $r \in \mathcal{C}$ then

$$\Pr_R [T^r[R] = \mathsf{accept}] \geq c.$$

Soundness For every $r \notin \mathcal{C}$ that is ϵ-far from $\mathcal{C}$

$$\Pr_R [T^r[R] = \mathsf{reject}] \geq s.$$

Strong Soundness For every $r \notin \mathcal{C}$

$$\Pr_R [T^r[R] = \mathsf{reject}] \geq s \cdot \delta_{\mathcal{C}}(r).$$

The parameters q, ϵ, s, c are known respectively as the *query complexity, distance threshold, soundness* and *completeness*.

When $c = 1$ we say the code and tester have *perfect completeness* and in such cases will often, for simplicity, omit reference to c.

Remark 1.2 (Distance threshold and high-error, or list-decoding, LTCs)**.** To get nontrivial LTCs the distance threshold ϵ should be less than half the relative distance of the code. Otherwise, it could be the case that there simply are no words ϵ-far from it, in which case the trivial tester that accepts all words shows that the code is $(0, \epsilon, 1, 1)$-LTC. We shall set the distance threshold to be one third the minimal distance of the code[1] and refer to such a $(q, \delta(\mathcal{C})/3, s, c)$-LTCs as a LTC for the *low-error regime*, or, simply, a *low-error* LTC. The choice of this name is because if $r \in \Sigma^n$ is accepted by the tester with probability greater than $1 - s$, we know that r is $\delta(\mathcal{C})/3$-close to $\mathcal{C}$, i.e., it has a *low fraction* of errors. Another common name for such a LTC is a *unique decoding LTC* because in the case just described there is a unique codeword that is closest to r.

For values of ϵ greater than half the minimal distance of $\mathcal{C}$, we say that $\mathcal{C}$ is a LTC in the *high-error*, or *list-decoding* regime. This is because a word accepted with probability greater than $1 - s$, which is known to be ϵ-close to $\mathcal{C}$, can in fact be ϵ-close to *list* of codewords. In the setting of high-error LTCs the kind of questions that are of interest revolve around understanding the connection between the acceptance probability of a received word and its proximity to the code. We shall not discuss these questions in this survey, due to scarcity of relevant results on rate limitations of such codes.

Remark 1.3 (Non-adaptivity and perfect completeness)**.** A tester is said to be *nonadaptive* if the codeword-entries queried by it depend only on the value of the random coins (in particular, they do not depend on answers given to earlier queries). All known LTC constructions are nonadaptive, i.e., the tester associated with them is nonadaptive. For a family of LTCs with perfect completeness and constant query complexity adaptivity can be assumed without loss of generality, by incurring at most a constant factor reduction in the soundness parameter. Furthermore, almost all known LTCs are linear and consequently can be assumed to be nonadaptive and with perfect completeness (cf. Theorem 2.4), the notable exception to both linearity and perfect completeness is the "long code" of Bellare et al. [1998].

Remark 1.4 (Soundness and completeness)**.** To get a meaningful definition we must require s to be greater than $1 - c$. Otherwise every code can be seen to be a $(0, 0, s, c)$-LTC, the tester associated with it rejects all words with probability s, hence accepts all words, and, in particular, all codewords, with probability $\geq c$.

[1] Some of the LTC rate limitations surveyed here, like [Ben-Sasson et al., 2003, Babai et al., 2005, Ben-Sasson et al., 2009], require the distance threshold to be less than one third the minimal distance. This is due to technical reasons arising in the proofs. In any case, all known LTC constructions work for any sufficiently small distance parameter and the standard assumption in property testing settings is that the distance threshold is an arbitrarily small nonzero constant.

Remark 1.5 (Running time)**.** Our definition of a tester does not put any limitation on the running time of the tester. For families of codes with constant query complexity this is not a severe restriction because the tester can always be assumed to run in (nonuniform) time that is at most polynomial in the blocklength, and under reasonable assumptions the running time is poly-logarithmic in the blocklength Meir [2008]. Families of linear codes — almost all known LTCs fall in this category — can be assumed to require (nonuniform) quasi-linear running time because they can be tested by "linear testers" (as explained in Section 2.1). The main advantage to not putting a running-time constraint on the tester is that it allows us to focus on the code structure and avoid questions about computational complexity.

1.2 A brief survey of known LTC constructions

The purpose of this section is to display the abundance and variance of LTC constructions which should motivate both the search for a common denominator to all the different ways LTCs are constructed, as well as the study of limitations of these codes.

LTCs based on low-degree polynomials The first family of locally testable codes, which was given by Blum, Luby, and Rubinfeld [1990], is the family of homomorphisms from a finite group G to a subgroup H of G. Formally, $\mathcal{C}(G,H) \subset H^G$ has one codeword corresponding to each group-homomorphism $\phi : G \to H$ and this codeword is the evaluation of ϕ on all elements of G. This family was shown to be a low-error LTC in [Blum et al., 1990]. The special case of G being the additive group $\mathbb{F}^n$ and $H = \mathbb{F}$ for a prime field $\mathbb{F}$ was shown by Bellare et al. [1996] to be locally testable in the high-error, or list-decoding, regime. The codes thus obtained are called Hadamard codes and correspond to the code of evaluations of n-variate, degree 1, homogenous polynomials. The generalization to arbitrary degree d polynomials was carried out promptly for the case of $d < |\mathbb{F}|$. This family of codes, known as Reed-Muller codes, was shown to be locally testable in the low-error regime in [Babai et al., 1991, Arora et al., 1998], and in the high-error regime by Raz and Safra [1997], Arora and Sudan [2003]. Later on the case of $d \geq |\mathbb{F}|$ was analyzed for the low-error regime by [Alon et al., 2005, Kaufman and Ron, 2006] and for the high-error regime by Samorodnitsky [2007] for the special case of $d = 2$. High-error LTCs based on polynomials of degree $d \geq 3$ and $d \geq |\mathbb{F}|$ remains as an interesting open problem.

Group invariant LTCs An "invariance-based" approach to the construction of LTCs was implicitly suggested by Alon et al. [2005] and explicitly undertaken, for the special case of affine-invariant codes, by Kaufman and Sudan [2008] (see also [Grigorescu et al., 2008, 2009b, Ben-Sasson and Sudan, 2010]). More on this approach can be found in Section 3 and in the survey of Sudan [2010]. Roughly speaking, this approach is based on finding codes that are invariant under a "sufficiently rich" group of permutations, and additionally contain some local constraints that all codewords satisfy. The group-invariance of the code then implies a multitude of local constraints that all codewords satisfy, and this leads the way to prove local-testability.

Composed LTCs Another way to construct LTCs, which among other things leads to the LTCs achieving the best known rate, relies on the use of probabilistically checkable proofs of proximity (PCPPs) [Ben-Sasson et al., 2006, Dinur and Reingold, 2006] (see also Meir [2009]). Another approach that is also described as "combinatorial", because it relies neither on properties of low-degree polynomials, nor on group theory, is based on taking a repeated tensor-product of codes

[Ben-Sasson and Sudan, 2006]. It should be pointed out that the codes arising from these methods are low-error LTCs and it remains to see what kind of LTCs in the high-error regime can emerge from high-soundness PCP composition techniques like those of [Moshkovitz and Raz, 2010, Dinur and Harsha, 2009].

Sparse unbiased LTCs The final family of LTCs we are aware of consists of *sparse, unbiased* binary linear codes, i.e., linear codes over $\mathbb{F}_p$ for prime p that have a number of codewords that is only polynomial in the blocklength and for which all nonzero codewords have relative weight that is very close to $1 - \frac{1}{p}$ [Kaufman and Sudan, 2007, Kopparty and Saraf, 2010] (see also Ben-Sasson and Viderman [2010a]).

1.3 Why study limitations of LTCs?

Before explaining why we think LTC limitations are worth pursuing we post the fundamental problem underlying our quest.

Question 1.6 (Do asymptotically good LTCs exist?)**.** Prove or refute the following statement: There exists an asymptotically good family of binary error correcting codes $\{\mathcal{C}_n \subseteq \{0,1\}^n \mid n \in \mathbb{Z}\}$ with relative distance δ that is a family of $(q, \delta/3, s, c)$-LTC, for some integer q and soundness and completeness parameters satisfying $c + s > 1$ (see Remark 1.4).

The main reason to study limits of LTCs is because this seems to be the most meaningful way to understand the limits of basic PCP-related parameters, most notably the *rate* of PCP proofs which we define as the ratio between the length of an **NP**-witness for an **NP**-instance ϕ , and the length of a probabilistically checkable proof for ϕ. The problem with the direct approach to bounding the rate of PCPs is that any nontrivial lower bound on the rate — even one that proves that PCP proof length is greater than zero — implies $\mathbf{P} \neq \mathbf{NP}$. Since all proofs of the PCP theorem make use of LTCs, and moreover the rate of the LTC is an upper bound on the rate of the PCP constructed from it, giving a negative answer to Question 1.6 would imply that PCP proofs constructed by current techniques will not attain constant rate. Anticipating future practical applications use of PCPs in cryptography and security-related protocols [Kilian, 1992, Micali, 2000, Barak and Goldreich, 2008], we see that understanding the rate of PCPs is very important not just for theoretical purposes.

More broadly, the study of limitations of locally testable codes can be viewed as a branch of the study of classical tradeoffs for error correcting codes. When new families of codes are discovered (e.g., linear, cyclic, maximal distance separable, algebraic geometry, turbo, etc.) it is of great importance to understand how well they match up with known codes in terms of their basic coding-related parameters. Locally testable codes possess a highly desirable coding-related property, namely, the amount of errors in a received word can be estimated by inspecting only a tiny fraction of the codeword. This leads to the possibility of saving computation time, and the energy consumption required by the decoding algorithm, by getting a quick and roughly accurate estimate of the condition of received words and asking for a "re-transmit" in case the word is estimated to be corrupted beyond repair.

Finally, the concept of "locality of computation" is a theme of great interest in numerous settings of theoretical computer science. This is witnessed by the large body of work on property testing and on locally decodable codes. Understanding the limits of LTCs also touches upon questions related to locality of computation in other settings and one may expect to see more connections between LTC rate bounds and other areas in which "local computation" is studied.

1.4 Summary of results appearing in the survey

In the next section we focus on linear codes and ask what limitations can be obtained from studying the structure of the set of dual codewords of small weight. We shall start with random low density parity check codes and use the expander-structure of the constraint graph associated with these codes to argue in Theorem 2.4 that they are not locally testable even when the query complexity is allowed to be fairly large. Then we shall generalize this result in Theorem 2.6 and show that all linear LTCs require that their dual code contain many low-weight words and, in Theorem 2.8, that these words must be nontrivially related. We conclude this section by showing in Theorem 2.11 that if an LTC has far too many redundant small-weight dual words then it has bad rate.

In Section 3 we shall investigate the rate limitations of group invariant codes. These codes include all known "base-case" LTCs, such as Hadamard and Reed-Muller codes, which serve as the building blocks in more elaborate LTC constructions (such as PCPP-based LTCs). We shall see in Theorem 3.3 that affine-invariant codes with small dual weight — the most general class of group-invariant codes known to be locally testable — has bad rate.

Results not covered by the survey Two lines of work on limits of LTCs are not surveyed here. The first set contains the results of Ben-Sasson et al. [2008] which show that 3-query LTCs arising from PCPP-based constructions cannot obtain close-to-optimal soundness in the list decoding regime without suffering a significant decrease in the code-rate. The second line discusses various kinds of 2-query LTCs — linear [Ben-Sasson et al., 2003], near-perfect completeness [Guruswami, 2006], "unique" [Kol and Raz, 2009a] and "affine" [Kol and Raz, 2009b] — and shows that there is at most a finite number of (2-query) LTCs of each kind.

2 Limiting rate of linear LTCs via the structure of the dual code

This section focuses on limitations on the rate of families of *linear* LTCs. We shall focus on the linear space that is dual to the (linear) code $\mathcal{C} \subseteq \mathbb{F}^n$, this space is also known as the *dual code* and defined as $\mathcal{C}^\perp = \{u \in \mathbb{F}^n \mid u \perp \mathcal{C}\}$ where $u \perp \mathcal{C}$ if and only if $u \perp w$ for all $w \in \mathcal{C}$ and $u \perp w$ denotes the equality $\sum_{i=1}^n u_i w_i = 0$ (in case of inequality we write $u \not\perp w$). We shall take particular interest in the combinatorial structure of the set of dual codewords of *small weight*. We start by explaining why focusing on this structure is all that matters for local testability of linear codes.

2.1 Linear LTCs are testable by linear testers

A natural way to test whether a word $r \in \mathbb{F}^n$ belongs to an $[n, k, d]_{\mathbb{F}}$ linear code $\mathcal{C} \subset \mathbb{F}^n$ is to project r onto a set of coordinates $I \subset \{1, \ldots, n\}, |I| \leq q$ and accept r if and only if this projection, denoted by $r|_I$, agrees with a projection $w|_I$ of some codeword $w \in \mathcal{C}$. Writing $\mathcal{C}|_I = \{w|_I \mid w \in \mathcal{C}\}$ we can describe this natural test as the test that accepts r if and only if $r|_I \in \mathcal{C}|_I$. The operator that projects $r \in \mathbb{F}^n$ onto I is a *linear* operator, by which we mean that for every $a, b \in \mathbb{F}^n$ and $\alpha, \beta \in \mathbb{F}$ we have $(\alpha a + \beta b)|_I = \alpha(a|_I) + \beta(b|_I)$ and this implies that our natural tester is fact a *linear test* — its acceptance predicate, defined as the subset of $\mathbb{F}^I$ of query-answer tuples accepted by the test, is a linear space, it is the precisely the linear space $\mathcal{C}|_I$.

Accordingly, a *linear tester* for $\mathcal{C}$ is given by a distribution D on subsets I of size at most q. The following theorem of Ben-Sasson et al. [2005] says that without loss of generality linear codes are q-query LTCs if and only if they are testable by a linear tester.

Theorem 2.1 (Linear LTCs have linear testers)**.** *If $\mathcal{C} \subseteq \mathbb{F}^n$ is a linear (q, ϵ, s, c)-LTC then $\mathcal{C}$ is a $(q, \epsilon, s + (1 - c), 1)$-LTC that can be tested by a linear tester. (Notice the difference between completeness and soundness is maintained when moving from an arbitrary tester to a linear one.)*

Given this theorem we can go one step further and describe the subsets $I \subset \{1, \ldots, n\}$ which correspond to nontrivial linear tests. If I is such that $\mathcal{C}|_I = \mathbb{F}^I$ then the (linear) test associated with I is meaningless — all words must be accepted by it. On the other hand if $\mathcal{C}|_I$ is a subspace strictly contained in $\mathbb{F}^I$ we do get a nontrivial test, meaning that some words $r \in \mathbb{F}^n \setminus \mathcal{C}$ will be rejected by it. In this case, the space that is dual to $\mathcal{C}|_I$, denoted $(\mathcal{C}|_I)^\perp$, has positive dimension, so it contains some nonzero words. Any word $u \in (\mathcal{C}|_I)^\perp$ can be extended to a word in $\mathbb{F}^n$ that is dual to $\mathcal{C}$ and has its nonzero entries contained in I — set all entries in $\{1, \ldots, n\} \setminus I$ to 0 and notice the word thus obtained is dual to $\mathcal{C}$.

Assuming $(\mathcal{C}|_I)^\perp$ is nontrivial we can think of another way to test wether $r|_I \in \mathcal{C}|_I$. Instead of querying all entries in I, pick a uniformly random $u \in (\mathcal{C}|_I)^\perp$ and accept r if and only if $u \perp r$. It is easy to see that this test retains perfect completeness, and we now argue that soundness goes down by a factor of at most $(1 - \frac{1}{\mathbb{F}})$. To see this, suppose $r \notin \mathcal{C}|_I$. The set $\left\{u \in (\mathcal{C}|_I)^\perp \;\middle|\; r \perp u\right\}$ is a strict subspace of $(\mathcal{C}|_I)^\perp$, hence it contains at most a $(1/|\mathbb{F}|)$-fraction of $(\mathcal{C}|_I)^\perp$, so a random $u \in (\mathcal{C}|_I)^\perp$ will "reject" r (i.e., $u \not\perp r$) with probability at least $(1 - 1/|\mathbb{F}|)$ times the probability that $r|_I \notin \mathcal{C}|_I$. To sum up, if we don't care too much about the exact soundness constant then we may assume without loss of generality that a linear LTC is tested by a tester that is defined by a distribution over $\mathcal{C}^\perp_{\leq q}$, the set of words in the dual code $\mathcal{C}^\perp$ that have weight at most q. We record this by the following corollary of Theorem 2.4 (cf. [Ben-Sasson et al., 2009, Section 2]). In what follows we use $u \sim D$ to denote that u is sampled according to the distribution D.

Corollary 2.2 (Linear codes are testable by a distribution over dual words of small weight)**.** *If $\mathcal{C} \subseteq \mathbb{F}^n$ is a linear (q, ϵ, s, c)-LTC then there exists a distribution D over $\mathcal{C}^\perp_{\leq q}$ such that for every r that is ϵ-far from $\mathcal{C}$ we have $\Pr_{u \sim D}[u \not\perp r] \geq s + (1 - c)(1 - 1/|\mathbb{F}|))$. (Notice the soundness is $(1 - 1/|\mathbb{F}|)$ times the soundness stated of Theorem 2.4.)*

All this leads us to consider the *constraint graph* of a tester, a concept that will play a pivotal role in our analysis. Given $U \subseteq \mathcal{C}^\perp_{\leq q}$ (U may be a strict subset of $\mathcal{C}^\perp_{\leq q}$) we define the constraint graph induced by U to be the bipartite graph $G(\{1, \ldots, n\}, U, E)$ with left vertex set $\{1, \ldots, n\}$, right vertex set U and an edge between i and u if and only if $u_i \neq 0$. Given a distribution D as in the corollary above let $\mathrm{supp}(D) = \left\{u \in \mathcal{C}^\perp \;\middle|\; D(u) > 0\right\}$ denote the support of the tester, it is the set of dual words, or linear tests, actually used by the tester. The constraint graph induced by a linear tester associated with D is the constraint graph induced by $\mathrm{supp}(D)$.

2.2 Random low density parity check codes

Roughly speaking, a linear code whose dual contains many small-weight words should be hard to construct as the existence of many small-weight words may reduce other parameters of the code, like its rate. Thus, a good starting point is to examine the local testability of the family of *random low density parity check* (LDPC) codes which are known to be asymptotically good [Gallager, 1962]. We shall show that testers achieving constant soundness for these codes require linear query complexity, and along the way we shall try to explain the way how this negative result about local testability is related to the structure of the constraint graphs associated with random LDPC codes.

To define our codes we need to describe the concept of a random regular bipartite graph. A bipartite graph is said to be (t, q)-*regular* if all vertices on the left side have degree at most t and

all vertices on the right side have degree at most q. A *random (t,q)-regular graph* with n left-hand vertices and $m = \lceil tn/q \rceil$ right-hand ones is obtained as follows. Start with a four-layered graph, the leftmost layer is V, the second and third have tn vertices each, numbered $1, \ldots, tn$, and the rightmost layer is U. Connect $i \in V$ to the t vertices in the second layer numbered $t(i-1)+1, \ldots, ti$. Similarly connect vertex number j in U to the q vertices numbered $q(j-1)+1, \ldots, qj$ in the third layer. (The m^{th} vertex may have less than q neighbors, in case tn/q is not an integer.) To obtain a *random* graph, pick a random permutation on tn elements and use it to construct a matching between the second and third layers. Finally, collapse each 3-edge-long path between $v \in V$ and $u \in U$ to obtain a single edge (collapsing parallel edges when needed), to obtain a random (t,q)-regular graph with n left vertices.

Definition 2.3 (Random low density parity check code)**.** The family of (t,q)-regular random LDPC codes is the distribution on families of linear codes obtained by picking the n^{th} member in the family according to the following process. For integers $t < q$ let $G = (V, U, E)$ be a random (t,q)-regular bipartite graph over n left vertices and $m = \lceil tn/q \rceil$ right vertices (notice $m < n$ because $t < q$). Associate each right-hand side vertex $\hat{u} \in U$ with the vector $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ defined by

$$u_i = \begin{cases} 1 & (i, \hat{u}) \in E \\ 0 & \text{otherwise.} \end{cases}$$

The LDPC code based on G is the code $\mathcal{C} = U^{\perp}$.

The rate of $\mathcal{C}$ is at least $\frac{n-m}{n} \approx 1 - \frac{t}{q}$ because $\dim(\mathcal{C}^{\perp}) \leq m$. It is well-known since the work of Gallager [1962] that a family of random LDPC codes is, with high probability, asymptotically good (cf. Sipser and Spielman [1996]). At first glance it may seem that such a family is locally testable. The set of q-query words U *characterizes* $\mathcal{C}$ by which we mean that $w \in \mathcal{C}$ if and only if $w \perp U$. And the random graph G is with high probability an expander which implies that for any set $S \subset \{1, \ldots, n\}, |S| = \epsilon n$ — think of S as indicating the minimal size set of bits that need be flipped in r to obtain a codeword — the set of indices of nonzero entries of a random $u \in U$ hits S with probability proportional to ϵ. In spite of all this $\mathcal{C}$ is not q-testable. This much was conjectured already by Spielman [1995]. Moreover, $\mathcal{C}$ is not even testable with any sublinear query complexity, i.e., a constant fraction of the received word must be queried in order to distinguish between completely uncorrupted, and severely corrupted, words. This is shown by the following theorem of Ben-Sasson et al. [2005].

Theorem 2.4 (Random LDPC codes require linear query complexity)**.** *For integers $t < q$ and constants $1/2 > \epsilon > 0, s > 0$ there exists $\mu > 0$ such that for sufficiently large n, a random (t,q)-LDPC code is, with high probability, not $(\mu n, \epsilon, s)$-locally testable.*

Proof Sketch. Consider a random LDPC code $\mathcal{C}$ based on a random (t,q)-regular graph G and assume that the constraints U that define it are linearly independent, which they are, with high probability. This linear independence implies that for every $u \in U$ there exists a word $r(u) \in \mathbb{F}_2^n$ such that

$$r(u) \not\perp u \quad \text{and} \quad r(u) \perp U \setminus \{u\}. \tag{1}$$

Appealing to the expansion properties of the graph G — which were used in the first place to argue that $\mathcal{C}$ has constant relative distance — we conclude that the code $\mathcal{C}_{-u} = (U \setminus \{u\})^{\perp} = \{w \mid w \perp (U \setminus \{u\})\}$ has good distance because the constraint graph induced by $U \setminus \{u\}$ is still a good expander. This implies that any word $r(u) \in \mathcal{C}_{-u} \setminus \mathcal{C}$ is ϵ-far from $\mathcal{C}$ for some constant $\epsilon > 0$.

What is the probability with which $r(u)$ is rejected by a q'-query tester? Recall that a linear q'-tester T is defined by a distribution D over $\mathcal{C}^{\perp}_{q'}$. Expressing a potential linear test $v \in \mathcal{C}^{\perp}_{q'}$ as a linear combination of elements from U and letting $U(v) \subseteq U$ denote the set of elements that have nonzero coefficients in this expression, we see from Equation (1) that $r(u) \not\perp v$ if and only if $u \in U(v)$. The answer to our question is then

$$\Pr_R[T^{r(u)}[R] = \mathsf{reject}] = \Pr_{v \sim D}[v \perp r(u)] = \Pr_{v \sim D}[u \in U(v)].$$

Taking one step further, for the tester defined by the distribution D to reject each $r(u)$ for $u \in U$, it better be the case that $U(v) \ni u$ for a random $v \sim D$ and uniformly random $u \in U$. This implies that a constant fraction of tests in $\mathrm{supp}(D)$ are, each, a linear combination of a constant fraction of U. Alas, with high probability, all words in $\mathrm{span}(U)$ that are a linear combination of a constant fraction of U must have large weight. This should sound reasonable because U is random, so summing up a constant fraction of its elements should result in a word with pretty large weight. We conclude that any tester that achieves constant soundness must be a distribution over words that have weight $\Omega(n)$, and this completes the proof (sketch). □

2.3 LTCs require redundant testers

Our next result rules out the existence of asymptotically good families of LTCs that lack sufficient *redundancy*, a concept we define next. This result can be seen as a generalization of the previous section to the case of codes that have "too few" dual words of weight q so let us explain how we quantify the number of such words and define what we mean by "too few" words.

If $\mathcal{C}^{\perp}_{\leq q}$ does not span all of $\mathcal{C}^{\perp}$ then $\mathcal{C}$ cannot be a q-query strong LTC because some non-codeword will be accepted with probability 1. This by itself does not yet mean that $\mathcal{C}$ is not locally testable, as it could be the case that all $r \notin \mathcal{C}$ that are accepted with probability 1 are, say, $(\epsilon/2)$-close to $\mathcal{C}$. A far more interesting case is when $\mathcal{C}^{\perp}_{\leq q}$ is a basis for $\mathcal{C}^{\perp}$ but contains no more words. Random (t, q)-regular codes give one example of such codes because it can be verified that the only words of weight at most q are those belonging to the linearly independent set U. We have already seen that such codes are not locally testable but perhaps other codes are? Before we continue let us formally define the redundancy of a code, which is the way we measure how many dual words are out there.

Definition 2.5 (Redundancy)**.** Given a set $U \subset \mathbb{F}^n$ let the *redundancy* of U be $\mathrm{redun}(U) = |U| - \dim(\mathrm{span}(U))$. It is the number of elements of U that can be removed from U without increasing the linear space that is dual to U (which we think of as a code $\mathcal{C}$). Notice $\mathrm{redun}(U) = 0$ if and only if U is linearly independent.

Let $\mathcal{C}$ be a $[n, k, d]_{\mathbb{F}}$-linear code. For D a distribution over $\mathcal{C}^{\perp}$ (think of D as a tester for $\mathcal{C}$) let $\mathrm{redun}(D) = \mathrm{redun}(\mathrm{supp}(D))$. D is said to be a *linearly independent tester* if $\mathrm{redun}(D) = 0$ and if moreover $\mathrm{supp}(D)$ spans $\mathcal{C}^{\perp}$ we call D a *basis tester* for $\mathcal{C}$. Finally, the q-redundancy of $\mathcal{C}$ is $\mathrm{redun}_q(\mathcal{C}) = \mathrm{redun}(\mathcal{C}^{\perp}_{\leq q})$.

The following theorem of Ben-Sasson et al. [2009] shows that any locally testable code with sufficiently large rate must be tested by redundant testers.

Theorem 2.6 (Linear LTCs require redundant testers)**.** *Let $\mathcal{C}$ be an $[n, k, d = \delta_0 n]_{\mathbb{F}}$-code that is a $(q, \delta_0/3, \epsilon)$-LTC. Then*

$$\mathrm{redun}_q(\mathcal{C}) \geq \frac{\epsilon k}{q} - 1.$$

Moreover, if D, the tester's distribution, is uniformly distributed over $\mathrm{supp}(D)$*, then*

$$\mathrm{redun}(D) \geq \frac{\epsilon - q/k}{1-\epsilon} \cdot (n-k).$$

The first equation above implies that every asymptotically good family of q-query LTCs must have linear q-redundancy, to see this set $k = \rho n$ where ρ is the rate of the family of codes. The second equation implies that q-query LTCs with super-constant size that are testable by a *uniform tester*, i.e., a tester whose distribution is uniform over a subset of $\mathcal{C}^{\perp}_{\leq q}$, must have linear redundancy. All algebraic and affine-invariant codes are testable by uniformly distributed testers, and so are sparse random unbiased codes but it should be stressed that the LTCs obtained by using composition techniques, such as PCPP-based and tensor-product ones, are not necessarily uniform. We point out that both inequalities are known to be nearly tight (cf. [Ben-Sasson and Viderman, 2010a]).

It may seem that the limitation placed by Theorem 2.6 on the minimal redundancy of an LTC can be easily overcome. Even if there are precisely $n-k$ linearly independent words in $\mathcal{C}^{\perp}_{\leq q}$ (this is what happens, for example, with random (t,q)-regular LDPC codes), there are $\binom{n-k}{2}$ words in $\mathcal{C}^{\perp}_{\leq 2q}$ — take the sumset of $\mathcal{C}^{\perp}_{\leq q}$ — so clearly this set has superlinear redundancy and for all we know $\mathcal{C}$ may be $2q$-testable without contradicting our theorem. The following stronger version of Theorem 2.6 is immune to the "sumset" trick and seems to say something deeper about the structure of small weight words of the dual code. To state this theorem we need a more refined definition of redundancy.

Definition 2.7 (Expected redundancy)**.** For $U \subset \mathbb{F}^n$, $B = \{b_1, \ldots, b_t\}$ a linearly independent set spanning $\mathrm{span}(U)$ (B is not necessarily a subset of U), and $u \in U$ let $B(u)$ be the set of elements of B used to represent u. If $u = \sum_{i=1}^{t} \beta_i b_i$ then this set is

$$B(u) = \{b_i \in B \mid \beta_i \neq 0\}.$$

For D a distribution on $\mathcal{C}^{\perp}_{\leq q}$ (which we view as a q-query tester for $\mathcal{C}$) let its *expected q-redundancy* be

$$\mathrm{Eredun}_q(D) = \min_B \mathbb{E}_{u \sim D}[|B(u)|]$$

where the minimum is taken over all bases $B \subset \mathcal{C}^{\perp}_{\leq q}$ which span $\mathcal{C}^{\perp}$. (Notice B is not necessarily a subset of $\mathrm{supp}(D)$.) The *expected q-redundancy* of $\mathcal{C}$, denoted as $\mathrm{Eredun}_q(\mathcal{C})$, is the minimal expected q-redundancy of a distribution D on $\mathcal{C}^{\perp}_{\leq q}$.

The following is the main theorem of Ben-Sasson et al. [2009].

Theorem 2.8 (LTCs require testers with large expected redundancy)**.** *Let $\mathcal{C}$ be an $[n, k, d = \delta_0 n]_{\mathbb{F}}$-code that is a $(q, \delta_0/3, s)$-LTC. Then*

$$\mathrm{Eredun}_q(\mathcal{C}) \geq \frac{sk}{q}.$$

Returning to the example discussed above, the example which assumed $\mathcal{C}^{\perp}_{\leq q}$ is linearly independent and suggested to use a $2q$-tester distributed over the sumset of $\mathcal{C}^{\perp}_{\leq q}$, it is not hard to see that its expected redundancy is 2 and to see this set $B = \mathcal{C}^{\perp}_{\leq q}$. Theorem 2.8 thus rules out this case, as well as that of taking as our tester any distribution over the $\Omega(k)$-wise sum of $\mathcal{C}^{\perp}_{\leq q}$.

Informally, this theorem says is that in order for a linear code to be q-query testable it must be the case that for any basis $B \subset \mathcal{C}^{\perp}_{\leq q}$ there exists a linear number of words in $\mathcal{C}^{\perp}_{\leq q} \setminus B$ that are each a linear combination of a constant fraction of B. This means that some nontrivial cancelation is going on by which many small-weight words — a linear number of them — are each a sum of many words from B.

2.4 Dense LTCs have small rate

In the previous section we saw that linear codes with too few dual words of small weight are not locally testable. In this section we discuss the opposite extreme, of codes with too many dual words of small weight. The following definition will be used to capture the notion of "too many" dual words.

Definition 2.9 (Dense codes)**.** An $[n, k, d]_{\mathbb{F}}$ linear code $\mathcal{C}$ is said to be (γ, q)-*dense* if for every $i \in \{1, \ldots, n\}$ there are at least γn^{q-2} dual words u of weight q such that $u_i \neq 0$.

For instance, the Hadamard code is $(\frac{1}{2}, 3)$-dense because every selection of $j \in \{1, \ldots, n\}$ participates in a dual word of weight 3 that touches i.

Remark 2.10. A different definition for dense codes can be suggested, one that uses the *total number* of weight-q dual words. For instance, we may decide to call a code $\mathcal{C}$ (γ, q)-dense' if $|\mathcal{C}^{\perp}_{\leq q}| \geq \gamma n^{q-1}$. This definition is problematic, as seen by taking the direct product of the Hadamard code with blocklength n, denoted H_n, with, say, a $[n, k = n/\mathrm{poly}\log n, d]_{\mathbb{F}_2}$-code $\mathcal{C}_0$ that is a $(3, \epsilon, s)$-LTC (codes with these parameters are known to exist). The resulting code

$$\mathcal{C} = \mathcal{C}_0 \times H_n = \left\{(c, c') \;\middle|\; c \in \mathcal{C}_0, c' \in H_n\right\}$$

is a linear 3-query LTC of blocklength $2n$ and can easily be seen to be $(1/4, 3)$-dense' because H_2 is $(1/2, 3)$-dense' but the rate of $\mathcal{C}$ is at least $k/2n$. In other words, we can artificially increase the density' of an LTC at the price of decreasing its rate by a constant factor.

It turns out that it is sufficient to consider the density of weight-3 and weight 4 words, due to the following claim because (γ, q)-density for $q \geq 3$ implies either $(3, \gamma')$- or $(4, \gamma')$-density for $\gamma' > 0$ depending only on γ. The main theorem of Ben-Sasson and Viderman [2010b] shows that dense codes have small rate:

Theorem 2.11 (Dense codes have small rate)**.** *For every $\gamma > 0$ and integer q there exists $\ell > 0$ depending only on γ and q such that the following holds. If $\mathcal{C}$ is a linear $[n, k, d]_{\mathbb{F}_2}$ code that is (γ, q)-dense, then the dimension k of $\mathcal{C}$ is at most $\log^{\ell} n$.*

The proof relies on results from additive combinatorics and we give a sketch of it next.

Proof Sketch. Take a generating matrix $A \in \mathbb{F}_2^{n \times k}$ for $\mathcal{C}$, a matrix satisfying $\mathcal{C} = \left\{Ax \;\middle|\; x \in \mathbb{F}_2^k\right\}$. Let $\mathcal{A} = \{A_i \mid i \in \{1, \ldots, n\}\} \subset \mathbb{F}_2^k$ denote the set of rows of the matrix. The density assumption implies

$$\Pr_{a, a' \in \mathcal{A}}[a + a' \in \mathcal{A}] \geq \gamma.$$

The Balog-Szemerédi-Gowers theorem [Balog and Szemerédi, 1994, Gowers, 1998], together with the Freiman-Ruzsa theorem [Freiman, 1973, Ruzsa, 1999], imply that $\mathcal{A}$ contains a subset $\mathcal{A}'$ of

size at least $\eta|\mathcal{A}|$ that is an η-fraction of some linear subspace of $\mathbb{F}_2^k$, where $\eta = \gamma^{\mathrm{poly}(1/\gamma)}$. In other words, the set of rows $\mathcal{A}'$ can be viewed, after an appropriate change of basis, as resulting from taking a constant fraction of the rows of a generating matrix of the Hadamard code, which is known to have very bad rate. Consider the residual set $\mathcal{A}'' = \mathcal{A} \setminus \mathcal{A}'$. The assumption that each $i \in \{1, \ldots, n\}$ touches many weight-3 words is now used to argue that $\mathcal{A}''$ is also $(\gamma', 3)$-dense, for $\gamma' > 0$ that depends only on γ, so our argument can be repeated. Continuing in this manner we reach the conclusion that the generating matrix A can be written, after a proper change of basis, as a block-diagonal matrix where each block is a constant fraction of a Hadamard code and Hadamard codes are known to have bad rate. Consequently, $\mathcal{C}$ has small rate and this completes the proof sketch. □

2.5 Question: Narrow the gap between redundant and dense LTC limitations

The rate limitations we have showed regarding both redundant, and dense, LTCs, suggest an interesting avenue for future research — to narrow the gap between these two cases. For simplicity consider the case of an asymptotically good family of *smooth* 3-query LTCs, i.e., LTCs that have a tester which queries each codeword entry with the same probability. The results on redundancy show that each member of the family should have at least a linear number of redundant weight-3 dual words. The result on dense codes shows that the overall number of such words is $o(n^2)$. Here is a seemingly simpler question that is currently open:

Question 2.12 (Number of small weight dual words of a linear LTC)**.** Prove or refute the following conjecture. Suppose $\{\mathcal{C}_n \subset \{0,1\}^n \mid n \in \mathbb{Z}\}$ is an asymptotically good family of linear $(3, \delta/3, s > 0)$-LTCs of relative distance at least $\delta > 0$. Suppose furthermore that $\mathcal{C}_n$ is testable by a tester associated with the uniform distribution on $(\mathcal{C}_n)^\perp_{\leq 3}$, the set of weight-3 words in $\mathcal{C}^\perp$. Then $|(\mathcal{C}_n)^\perp_{\leq 3}| = \omega(n)$.

3 Limitations on group-invariant codes

We have seen in Section 2.3 that linear LTCs must have dual codes whose small-weight words show a large degree of nontrivial redundancy. Constructing codes that have large rate and such a level of redundancy seems like a hard problem, and one way to get around it is to use codes that are invariant under a "sufficiently rich" group (a concept we explain next), for which the existence of even a single small-weight dual word immediately implies a large number of such words.

A code $\mathcal{C}$ of blocklength n induces a group of automorphisms $\mathrm{aut}(\mathcal{C})$, this is the group of permutations $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ under which the code is invariant, by which we mean that for every $w = (w_1, \ldots, w_n) \in \mathcal{C}$ the π-permuted word $\pi(w) = (w_{\pi(1)}, \ldots, w_{\pi(n)})$ also belongs to $\mathcal{C}$. It is not hard to verify that $\mathrm{aut}(\mathcal{C})$ is indeed a group and that $\mathrm{aut}(\mathcal{C}^\perp) = \mathrm{aut}(\mathcal{C})$. Consequently, if $\mathcal{C}^\perp$ contains a word u of weight q then $\mathcal{C}^\perp_{\leq q}$ contains $\{\pi(u) \mid \pi \in \mathrm{aut}(\mathcal{C})\}$. Thus, if $\mathrm{aut}(\mathcal{C})$ is sufficiently rich we can hope for $\mathcal{C}^\perp_{\leq q}$ to be large and redundant and, if all stars align properly, $\mathcal{C}$ will be a q-query LTC and moreover have large rate and relative distance.

Two notable families of groups that should be mentioned in this context are doubly transitive and affine-invariant ones. A group of permutations G over n elements is said to be *doubly transitive*, or 2-*wise transitive*, if for every $i \neq j$ and $i' \neq j' \in \{1, \ldots, n\}$ there exists $\pi \in G$ such that $\pi(i) = i'$ and $\pi(j) = j'$. A conjecture attributed[2] to Alon et al. [2005] is that all codes which

[2]We use the term "attributed" because in [Alon et al., 2005, Section 5] it appears as an open question.

are invariant under a doubly transitive group (call them doubly transitive codes) are testable with query complexity q' that depends only on the smallest q for which $\mathcal{C}^{\perp}_{\leq q}$ spans $\mathcal{C}^{\perp}$. In particular, this query complexity is conjectured to be independent of the blocklength of $\mathcal{C}$. (The requirement that $\mathcal{C}^{\perp}_{\leq q}$ span $\mathcal{C}^{\perp}$, cannot be replaced by the weaker assumption that q is the minimal distance of $\mathcal{C}^{\perp}$. Grigorescu et al. [2009a] showed that if one opts for the weaker assumption then the conjecture is false.) It is shown by Kaufman and Viderman [2010] that doubly transitive codes with small dual distance are so-called locally correctable codes. These codes are a stronger analog of locally decodable codes (cf. [Goldreich, 2005, Trevisan, 2004]), and this implies a polynomial upper bound on their rate of the form of the form $\rho(\mathcal{C}) = O\left(\log n \left(\frac{\log n}{n}\right)^{\frac{2}{q+1}}\right)$, as shown by Woodruff [2007]. This raises the following open problem:

Question 3.1 (Polynomial rate doubly transitive LTCs)**.** Does there exist a family $\{\mathcal{C}_n \subseteq \mathbb{F}^n \mid n \in \mathbb{Z}\}$ of doubly transitive $(q, \epsilon > 0, s, 1)$-LTCs that has inverse polynomial rate, i.e., $\rho(\mathcal{C}_n) \geq 1/n^{O(1)}$?

A group is said to be *affine-invariant* if $\{1, \ldots, n\}$ can be identified with a vector space $\mathbb{K}^m$ over a finite field $\mathbb{K}$ and G is then isomorphic to the group of invertible affine transformations[3] over $\mathbb{K}^m$. The family of affine-invariant codes includes the Hadamard and Reed-Muller codes as well as dual-BCH codes. Kaufman and Sudan [2008] showed that, when $|\mathbb{K}|$ is small, every affine-invariant family of codes over $\mathbb{K}^m$, whose dual contains a small-weight word, is locally testable. Since every affine group is doubly transitive, the work of Kaufman and Sudan [2008] shows that the double-transitivity conjecture does hold in certain interesting special cases. Later on we shall see that affine-invariant codes have small rate, and this answers negatively the question above for this special case.

A third and final family of group invariant codes considered in the literature is that of *cyclic* codes, i.e., codes invariant under a cyclic group. All affine invariant codes (including Hadamard and Reed-Muller) are, in particular, cyclic. Babai et al. [2005] showed that a family of cyclic LTCs cannot be asymptotically good, either its rate or its distance must be less than $1/\sqrt{\log n \log\log n}$. A long-standing open problem in coding theory is whether there exists an asymptotically good family of cyclic codes (cf. [MacWilliams and Sloane, 1978, Open Problem 9.2]). The result above shows that when local testability is thrown in as a requirement, then indeed asymptotically good codes do not exist.

3.1 Affine invariant LTCs have small rate

In this section we discuss rate limitations of affine-invariant locally testable codes. More information on this topic can be found in the survey of Sudan [2010]. Recall that if $\mathcal{C}$ is an $[n, k, d]_{\mathbb{F}}$-code affine-invariant code it means we can identify $\{1, \ldots, n\}$ with $\mathbb{K}^m$ for some field $\mathbb{K}$ which is a finite extension[4] of $\mathbb{F}$ and such that the automorphism group of $\mathcal{C}$ contains the affine (semi-)group over $\mathbb{K}^m$. The study of affine invariant LTCs was initiated by Kaufman and Sudan [2008], as a first step towards characterizing the class of "algebraic" properties which are testable. This class is also

[3] The work of Kaufman and Sudan [2008] actually talks about the semi-group of all affine transformations, including the non-invertible ones.

[4] The more general case of $\mathbb{K}$ being an arbitrary field, not necessarily extending $\mathbb{F}$, has not been addressed so far. However, it seems reasonable to expect that such codes should not have good rate, regardless of their local testability properties. This is because $\mathbb{K}^m$-affine invariance and $\mathbb{F}$-linearity do not mix well when $\mathbb{K}$ is not an extension of $\mathbb{F}$.

an interesting special case of the doubly transitive conjecture of Alon et al. [2005]. Indeed, for such codes Kaufman and Sudan [2008] showed that local testability exists as long as the field $\mathbb{K}$ is sufficiently small and the dual code has constant distance, as seen from their main theorem:

Theorem 3.2 (Affine invariant codes over small fields with constant dual distance are locally testable)**.** *For fields $\mathbb{F} \subseteq \mathbb{K}$ let $\mathcal{C}$ be an $[n = |\mathbb{K}^m|, k, d]_{\mathbb{F}}$ affine-invariant code such that $\mathcal{C}^{\perp}$ contains a word of weight q_0. Then $\mathcal{C}$ is*

$$\left(q = (|\mathbb{K}|^2 q_0)^{|\mathbb{K}|^2}, s = \frac{1}{2(2q+1)(q+1)}\right) \textit{-strongly locally testable}$$

by which we mean that there exists a q-query linear tester that rejects noncodewords $r \notin \mathcal{C}$ with probability at least $s \cdot \delta_{\mathcal{C}}(r)$.

Now we discuss the rate of such codes. Since affine invariant codes are cyclic, one could get an inverse logarithmic bound on the rate of affine invariant LTCs from what is known on cyclic LTCs. A tighter, inverse polynomial, bound on the rate follows from the result of Kaufman and Viderman [2010] which says that such codes are locally decodable (and locally correctable) and the result of Woodruff [2007] which bounds the rate of locally decodable codes by $O\left(\log n \left(\frac{\log n}{n}\right)^{\frac{2}{q+1}}\right)$. The following result of Ben-Sasson and Sudan [2010] gives a stronger bound, showing that the dimension of affine-invariant codes is merely polylogarithmic in the blocklength of the code.

Theorem 3.3 (Affine invariant LTCs have small rate)**.** *Let p be a prime and r, n, m be positive integers and let $\mathbb{F}$ be the field of size p^r and $\mathbb{K}$ be its degree ℓ-extension, which is of size $p^{r\ell}$. Any affine invariant $[n = |\mathbb{K}^m|, k, d]_{\mathbb{F}}$-code $\mathcal{C}$ such that $\mathcal{C}^{\perp}$ contains a word of weight $q > 0$ satisfies*

$$k \leq (\log_p n)^{q-1}.$$

Notice the theorem shows exponential rate even for large fields $\mathbb{K}$, which are not known to be locally testable. We point out that the theorem as stated in [Ben-Sasson and Sudan, 2010] gives more information on affine-invariant codes with small dual distance, showing they are subcodes of low-degree polynomials (Reed-Muller codes). We shall not describe this result, nor shall we go into details of the proof because quite a lot of algebra is needed to describe it. Instead, we point the interested reader to the survey [Sudan, 2010] and the relevant papers [Kaufman and Sudan, 2008, Ben-Sasson and Sudan, 2010].

We end this section by pointing out the following interesting question which addresses the rate of a natural family of codes invariant under a linear group (in particular, Theorem 3.3 does not apply to such codes):

Question 3.4 (Rate of linear invariant codes with small dual distance)**.** Let $\mathbb{K}$ be a finite extension of a finite field $\mathbb{F}$. Let $GL(m, \mathbb{K})$ denote the general linear group over $\mathbb{K}$, containing all invertible m-dimensional linear transformations over $\mathbb{K}$. Let $\mathcal{C}$ be an $[n = |\mathbb{K}|^m, k, d]_{\mathbb{F}}$-linear code that is invariant under $GL(m, \mathbb{K})$ and suppose $\mathcal{C}^{\perp}$ contains a word of weight $q > 0$. How large can k be as a function of the field size $|\mathbb{K}|$ and code distance d?

Acknowledgement

Thanks to Michael Viderman for helpful comments on an earlier draft. The research leading to some of the results surveyed here has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258 and from the US-Israel Binational Science Foundation under grant number 2006104.

References

Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

S. Arora and M. Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3): 365–426, 2003.

Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

L. Babai, L. Fortnow, L.A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32. ACM, 1991.

L. Babai, A. Shpilka, and D. Stefankovic. Locally testable cyclic codes. *IEEE Transactions on Information Theory*, 51(8):2849–2858, 2005.

Antal Balog and Endre Szemerédi. A statistical theorem of set addition. *Combinatorica*, 14(3): 263–268, 1994.

Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM J. Comput.*, 38(5):1661–1694, 2008.

M. Bellare, D. Coppersmith, J. Hastad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.

Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Struct. Algorithms*, 28(4):387–402, 2006.

Eli Ben-Sasson and Madhu Sudan. Limits on the rate of locally testable affine-invariant codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (108), 2010.

Eli Ben-Sasson and Michael Viderman. Low rate is insufficient for local testability. In Ronen Shaltiel, editor, *Proc. 14th Intl. Workshop on Randomization and Computation - RANDOM 2010*, Sept. 2010a.

Eli Ben-Sasson and Michael Viderman. Dense locally testable codes have bad rate. Unpublished manuscript, 2010b.

Eli Ben-Sasson, Oded Goldreich, and Madhu Sudan. Bounds on 2-query codeword testing. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *RANDOM-APPROX*, volume 2764 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2003. ISBN 3-540-40770-7. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2764&spage=216`.

Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM J. Comput*, 35(1):1–21, 2005. URL `http://epubs.siam.org/SICOMP/volume-35/art_44544.html`.

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.

Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-query PCPPs are long. In *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2008. ISBN 978-3-540-70574-1. URL `http://dx.doi.org/10.1007/978-3-540-70575-8_56`.

Eli Ben-Sasson, Venkatesan Guruswami, Tali Kaufman, Madhu Sudan, and Michael Viderman. Locally testable codes require redundant testers. In *CCC '09: Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, pages 52–61, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3717-7. doi: http://dx.doi.org/10.1109/CCC.2009.6.

Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *STOC*, pages 73–83. ACM, 1990.

Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12:1–12:44, June 2007. ISSN 0004-5411.

Irit Dinur and Prahladh Harsha. Composition of low-error 2-query pcps using decodable pcps. In *FOCS*, pages 472–481. IEEE Computer Society, 2009. ISBN 978-0-7695-3850-1.

Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput*, 36(4):975–1024, 2006. URL `http://dx.doi.org/10.1137/S0097539705446962`.

G. A. Freiman. *Foundations of a structural theory of set addition*, volume 37. American Mathematical Society, 1973.

R. Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1): 21–28, 1962.

Oded Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005. URL `http://eccc.hpi-web.de/eccc-reports/2005/TR05-014/index.html`.

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

W. T. Gowers. A new proof of szemerèdi's theorem for arithmetic progressions of length four. *Geom. Funct. Anal.*, 8(3):529–551, 1998.

E. Grigorescu, T. Kaufman, and M. Sudan. Succinct representation of codes with applications to testing. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 534–547, 2009a.

Elena Grigorescu, Tali Kaufman, and Madhu Sudan. 2-transitivity is insufficient for local testability. In *IEEE Conference on Computational Complexity*, pages 259–267. IEEE Computer Society, 2008. ISBN 978-0-7695-3169-4. URL `http://doi.ieeecomputersociety.org/10.1109/CCC.2008.31`.

Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 5687 of *Lecture Notes in Computer Science*, pages 534–547. Springer, 2009b. ISBN 978-3-642-03684-2.

Venkatesan Guruswami. On 2-query codeword testing with near-perfect completeness. In Tetsuo Asano, editor, *ISAAC*, volume 4288 of *Lecture Notes in Computer Science*, pages 267–276. Springer, 2006. ISBN 3-540-49694-7. URL `http://dx.doi.org/10.1007/11940128_28`.

Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput.*, 36(3): 779–802, 2006.

Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.

Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 403–412. ACM, 2008. ISBN 978-1-60558-047-0. URL `http://doi.acm.org/10.1145/1374376.1374434`.

Tali Kaufman and Michael Viderman. Locally testable vs. locally decodable codes. In Ronen Shaltiel, editor, *Proc. 14th Intl. Workshop on Randomization and Computation - RANDOM 2010*, Sept. 2010.

Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732. ACM, 1992.

G. Kol and R. Raz. Bounds on 2-Query Locally Testable Codes with Affine Tests. *ECCC Report TR09-138*, 2009a.

G. Kol and R. Raz. Locally testable codes analogues to the unique games conjecture do not exist. *ECCC Report TR09-128*, 2009b.

Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. In Leonard J. Schulman, editor, *STOC*, pages 417–426. ACM, 2010. ISBN 978-1-4503-0050-6.

F.J. MacWilliams and N.J.A. Sloane. *The theory of error-correcting codes*. North-Holland Amsterdam, 1978.

Or Meir. On the efficiency of non-uniform pcpp verifiers. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(064), 2008.

Or Meir. Combinatorial construction of locally testable codes. *SIAM J. Comput.*, 39(2):491–544, 2009.

Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.

Dana Moshkovitz and Ran Raz. Two-query pcp with subconstant error. *J. ACM*, 57(5), 2010.

R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 1997.

Imre Z. Ruzsa. An analog of freiman's theorem in groups. *Astèrique*, 258:323–326, 1999.

Alex Samorodnitsky. Low-degree tests at large distances. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 506–515. ACM, 2007. ISBN 978-1-59593-631-8.

M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6): 1710–1722, 1996.

D.A. Spielman. *Computationally efficient error-correcting codes and holographic proofs*. PhD thesis, MIT, 1995.

M. Sudan. Invariance in Property Testing. *ECCC, TR10-051*, 2010.

Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004.

D. Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(006), 2007.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de