TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
The Raymond and Beverly Sackler Faculty of Exact Sciences
The Blavatnik School of Computer Science

A Theory of Locally Decodable Codes

Thesis submitted for the degree of Doctor of Philosophy
by
Klim Efremenko

This work was carried out under the supervision of
Professor Oded Regev and Professor Amnon Ta-Shma

Submitted to the Senate of Tel Aviv University
August 2012

This work is dedicated to my wife Rivka and to my children Racheli Bracha, Matan and Shlomo.

Acknowledgements

I would like to thank my advisors Oded Regev and Amnon Ta-Shma for their sage advise and a lot of support during my PhD. I would like to thank them for caring so much and for helping me with everything that I have ever asked for and even more.

I would like to thank my lab mates, Avi Ben-Aroya, Iftah Gamzu, Michal Moshkovitz, and Ishay Haviv for many interesting conversations and discussions.

I also thank the Israel Science Foundation and the Adams Fellowship Program of the Israel Academy of Sciences and Humanities for its generous financial support.

I would like to thank Ely Porat and Omer Reingold, my advisors during my work on Chapter 2. I would also like to thank to Irit Dinur for introducing to me the area of locally decodable codes and for her help during the work on the results in the second chapter.

I would also like thank to Ariel Gabizon, Oded Goldreich, Dmitry Gourevitch, Venkatesan Guruswami, Shachar Lovett, David Woodruff, Alex Lubotzky, Zeev Rudnik, Avi Wigderson, Chris Umans, Or Meir, Danny Gutfreund and Sergey Yekhanin for their meaningful discussions.

I would like thank to Tel-Aviv University for providing a research environment. I would like thank to Weizmann Institute and Bar-Ilan University. Most of the results in Chapter 2 where obtained while I was working at these universities.

I am thankful to my parents, who placed me on the beginning of the academic journey so naturally, and who keep supporting me on this road and for always being proud of my achievements.

But most of all I wish to thank my wife Rivka and my children who were born during my PhD, Racheli Bracha, Matan and Shlomo, for their love, patience, dedication, encouragement and support on this long, and sometimes frustrating, adventure. I lovingly dedicate this thesis to them.

Abstract

In this thesis we study Locally Decodable Codes. A code $\mathcal{C}$ 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 $\mathcal{C}(x)$, such that even if a constant fraction of $\mathcal{C}(x)$ is corrupted, the decoding algorithm returns the correct answer with high probability.

LDCs are important not because of their obvious applications to data transmission and data storage but because of their 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. Locally Decodable Codes also found applications in data structures and fault tolerant computations.

Locally Decodable Codes implicitly appeared in the PCP literature already in the early 1990s, most notably in [2, 37]. However the first formal definition of LDCs was given by Katz and Trevisan [29] in 2000. Since then LDCs became widely used. The first constructions of LDCs [4, 29] were based on polynomial interpolation techniques. Later on more complicated recursive constructions were discovered [5, 43]. All these constructions had exponential length. The tight lower bound of $2^{\Theta(n)}$ codes were given in [19, 32] for two queries LDCs. For many years it was conjectured (see [16, 17]) that LDCs should have an exponential dependence on n for any constant number of queries, until Yekhanin's breakthrough [45]. Yekhanin obtained 3-query LDCs with sub-exponential length. Yekhanin's construction is based on an unproven but a highly believable conjecture in number theory.

Our Results In this thesis we obtain the following results:

- In Chapter 2 we define a framework of matching vector codes which gives the first unconditional construction of sub-exponential locally decodable codes. Formally S-matching vectors are two sets $\{u_1, \ldots, u_k\}$ and $\{v_1, \ldots, v_k\}$ such that $\langle u_i, v_i \rangle = 0$ and $\langle u_i, v_j \rangle \in S$ for $i \neq j$. We show how from matching vectors and a S-decoding polynomial one can construct LDCs. Using this construction together with Grolmusz's [23] construction of matching vectors, we obtain the first unconditional sub-exponential LDCs.

  Most of the work on this chapter was done while I was in Bar-Ilan University and Weizmann Institute under the supervision of Ely Porat and Omer Reingold . This chapter is based on paper [12].

- In Chapter 3 we show the connection between locally decodable codes and irreducible representations. More precisely we show that if there exists an irreducible representation (ρ, V) of G and q elements $g_1, g_2, \ldots, g_q$ in G such that there exists a linear combination of matrices $\rho(g_i)$ that is of rank one, then we can construct a q-query Locally Decodable Code $\mathcal{C} : V \to \mathbb{F}^G$.

We show that this approach captures sub-exponential constructions of MVC and Reed-Muller codes.

This chapter is based on paper [13].

- In Chapter 4 we show how to amplify error-tolerance of locally decodable codes. Specifically, this shows how to transform a locally decodable code that can tolerate a constant fraction of errors to a locally decodable code that can recover from a much higher error-rate, and how to transform such locally decodable codes to *locally list-decodable codes*. This chapter is based on the paper [7].

Contents

1 Introduction **1**

1.1 Matching Vector Codes . . . 3
1.2 LDCs from irreducible representations . . . 3
1.3 Noise tolerance of LDCs . . . 5
1.4 Collaborators . . . 7

2 Matching Vector Codes **8**

2.1 Definitions and Basic Facts . . . 8
2.2 Locally Decodable Codes . . . 9
2.2.1 Matching Sets of Vectors . . . 9
2.2.2 S-Decoding Polynomials . . . 10
2.2.3 Matching Vector Codes . . . 11
2.3 A Simple Construction of S-Matching Vectors . . . 13
2.4 Binary Locally Decodable Codes . . . 15
2.5 Future Work . . . 17

3 Locally Decodable Codes from Irreducible Representations **18**

3.1 Notation and Preliminaries . . . 18
3.1.1 Representation Theory . . . 18
3.1.2 Locally Decodable Codes . . . 22
3.2 Locally Decodable Codes from Irreducible Representations . . . 22
3.2.1 Example: Two Query LDC from Representations of S_n . . . 24
3.2.2 Embedding to the Regular Representation . . . 25
3.2.3 Alphabet Reduction . . . 27
3.3 Matching Vector Codes and Abelian Invariant Codes . . . 28
3.4 Is Irreducibility Essential? . . . 30
3.4.1 Yes! . . . 31
3.4.2 No! . . . 31
3.5 G-Invariant Codes and Representations of G . . . 33

4 Amplifying the Error-Tolerance of Locally Decodable Codes **35**

4.1 Definitions . . . 35
4.2 Composition Theorem . . . 37

5 Open Problems **39**
5.1 Locally Decodable Codes 39
5.2 Self Correctable Codes 40

Bibliography **41**

Chapter 1

Introduction

Everything New Is Actually Well-Forgotten Old

In this thesis we study *Locally Decodable Codes*. A Locally Decodable Codes is a code that allows the retrieval of any symbol of a message by reading only a constant number of symbols from its codeword, even if a large fraction of the codeword is adversarially corrupted. Formally, a code $\mathcal{C}$ is said to be locally decodable with parameters (q, δ, ϵ) if for every message x and for all indices i it is possible to recover any symbol x_i of message x by making at most q queries to $\mathcal{C}(x)$, such that even if a δ fraction of $\mathcal{C}(x)$ is adversarially corrupted, the decoding algorithm returns the correct answer with probability at least $1 - \epsilon$.

LDCs have an obvious application for data transmission and data storage. However the importance of LDCs comes not from these applications but from their applications in theoretical computer science and cryptography. LDCs were first used in the context of worst case – average case reductions and probabilistically checkable proofs. Later LDCs and their variations found their applications in many important results in pseudo-random generators, hardness amplification, private information retrieval schemes. See the surveys [16, 40] for more details.

Locally decodable codes were first formally defined by Katz and Trevisan [29], although this notion already appeared implicitly in previous works. When the number of queries is $poly \log k$, where k is the length of the message, Reed-Muller codes give polynomial length LDCs. When the number of queries is k^{ε}, a variant of Reed-Muller [34] gives LDCs of rate approaching one. For a detailed survey on recent results in LDCs see the survey by Yekhanin [46]. We summarize the current results of lower and upper bounds in the Table 1.1

Hadamard and Reed-Muller Codes Before continuing the discussion on modern constructions of LDCs, let us give two classical examples of LDCs: Hadamard codes and Reed-Muller codes. The Hadamard code encodes k bits to 2^k bits where each coordinate of the message corresponds to some value in $\{0,1\}^k$. We encode a message $m = (m_1, m_2, \ldots, m_k)$ to the inner product of this message with all coordinates, i.e., at the coordinate $x \in \{0,1\}^k$ we write $\langle m, x \rangle \triangleq \sum m_i x_i$. In order to decode the ith bit of the message we pick a random $y \in \{0,1\}^k$ and return the xor of the coordinates y and $y + e_i$ (where e_i is an element with 1 at ith coordinate and zero elsewhere). Note that if our codeword is corrupted in at most $\delta < \frac{1}{4}$ fraction of coordinates, then with probability at least $1 - 2\delta$ both coordinates y and $y + e_i$ will not be corrupted. In this case we have calculated $\langle m, y \rangle + \langle m, y + e_i \rangle = \langle m, e_i \rangle = m_i$. Thus we will return the correct answer with probability at least $1 - 2\delta$.

q	Lower Bounds	Upper Bounds
1	Do not exist [KT]	
2	2^n [GKST,KdW]	2^n (Hadamard)
3	$\Omega(n^{3/2})$[KT] $\Omega(\frac{n^2}{\log^2 n})$[KdW] $\Omega(\frac{n^2}{\log n})$ [W]	$\exp(n^{\varepsilon})$ [Y] $\exp(\exp(\frac{\log n}{\log\log n}))$*[Y] *under number theoretic conjecture $\exp(\exp(O(\sqrt{\log n \log\log n})))$
> 3	$\Omega\left(\frac{n^{1+1/(\lceil q/2\rceil-1)}}{\log n}\right)$	$\exp(\exp(O(\sqrt[\log q]{\log n \log\log n})))$
polylog(n)		poly(n)
n^{ε}		$(1+\varepsilon)n$ [KSY]

Table 1.1: Lower and Upper Bounds on LDCs.

The second example is the Reed-Muller codes. The code has two parameters (d, n) and it is defined over field $|\mathbb{F}| > d$. The code sends a polynomial in n variables of total degree d to its evaluation on all points in $\mathbb{F}^n$. Now let us show that a Reed-Muller code is a $d+1$-query LDC. We make even a stronger claim: Reed-Muller is a self correctable code. A q-query self correctable code is a code such that any coordinate of the code could be corrected with high probability by reading at most q coordinates of the code even if a constant fraction of the coordinates is corrupted. Note that linear self correctable codes are also LDCs since one can always encode the message in the coordinates of the code. Now let us show that the Reed-Muller code is self correctable. Note that if we restrict any multivariate polynomial $p(\vec{x})$ of total degree d to any line $\ell = \{\vec{a}t + b\}$ by $p_\ell(t) = p(\vec{a}t + b)$ then we will get a polynomial of total degree at most d. Assume that we want to recover a value of the code at point $\vec{b}$. Pick at random $0 \neq \vec{a} \in \mathbb{F}^n$ and query the code at random $d+1$ points on the line $\ell = \{\vec{a}t + \vec{b}\}, t \neq 0$. From these points reconstruct the polynomial p_ℓ of degree d. The value of this polynomial at zero is $p_\ell(0) = p(\vec{b})$. Note that since we pick $\vec{a}$ at random, each query is uniformly distributed. Thus, if our code was corrupted in at most δ fraction of coordinates, all of our queries will be uncorrupted with probability at least $1 - (d+1)\delta$.

The Hadamard code is an example of a 2-query LDC of exponential length. A tight lower bound of $2^{\Theta(k)}$ on the length of *linear* 2-query locally decodable codes was given in Goldreich et al. [18] and was extended to general codes by Kerenidis and de Wolf [31]. When the number of queries is constant and greater than two, much less is known. For an arbitrary constant number of queries q, only weak super-linear lower bounds are known, see [29, 31, 41].

For many years it was conjectured (see [16, 17]) that LDCs should have an exponential dependence on n for any constant number of queries, until Yekhanin's breakthrough [45]. Yekhanin obtained 3-query LDCs with sub-exponential length. Yekhanin's construction is based on an unproven but a highly believable conjecture in number theory that there are infinitely many Mersenne primes.

The goal of this thesis is to understand what will allow us to construct good LDCs. We will present two frameworks for the construction of LDCs:

1.1 Matching Vector Codes

In [12] we develop a combinatorial framework *matching vector codes* (MVC) for a construction of LDCs. The heart of this framework is a combinatorial object: matching vectors which fortunately for us were highly investigated in combinatorics. This framework is a kind of a generalization and simplification of Yekhanin's result [45]. We extend Yekhanin's construction to work not only with prime but also with composite numbers. Fortunately Grolmusz [23] had showed that there exists constructions of matching vectors over composite numbers with much better parameters than over prime numbers. Using these matching vectors allows us to give an *unconditional* construction of sub-exponential LDCs. LDCs implied from this framework also have much better parameters. Let us summarize this in the following theorem:

Theorem 1.1.1 ([12]). *For every r and for every k there exists $q \leq 2^r$ query LDC $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^n$, where $n = \exp(\exp(O(\sqrt[r]{\log n (\log\log n)^{r-1}})))$.*

Number of queries in this theorem depends on the object called S-decoding polynomial. A trivial construction of S-decoding polynomial leads to the bound 2^r on number of queries. When $r = 2$ we also show how to reduce the number of queries from $2^2 = 4$ to 3. Later in [28, 35] this construction was extended (under some number theoretic assumptions) to any r. This reduced number of queries to $3^{r/2}$. Limits of this framework were also studied in the consequence works [9, 11], where it was shown that for a restricted framework of MVC is impossible to construct polynomial LDCs.

Although significant progress was made in understanding LDCs, the gap between lower and upper bounds is still very large. While lower bounds are only slightly more than linear, upper bounds are only sub-exponential. Today all known sub-exponential constructions of LDCs with constant number of queries could be described in the framework of *matching vector codes* (MVCs). It seems that in order to make a significant improvement in MVCs, we need to improve matching vectors, where there was almost no progress in the last ten years. 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 [23] breakthrough. This construction is the basis for our subexponential constructions of LDCs. For both MVs and LDCs, there is not even a conjecture today of what are their best possible parameters.

1.2 LDCs from irreducible representations

Our second framework for construction MVCs is based on representation theory. Although the framework of MVCs is pretty simple, it still does not explain the real nature of LDCs. This leads us to seek a new approach to understanding LDCs, which in turn leads us to start a systematic study of LDCs from the point of view of representation theory. We present another framework for the construction of LDCs and show that it captures two important classes of LDCs: Reed Muller codes and MVCs. We believe that this is the real algebraic nature behind LDCs.

Let us describe this framework in more details. Let G be a finite group. A *representation* of the group G is a pair (ρ, V) of a vector space V and a mapping $\rho : G \to GL(V)$ from G to the group of invertible matrices over V which is a group homomorphism, i.e., for all $g_1, g_2 \in G$ it

holds that $\rho(g_1 g_2) = \rho(g_1)\rho(g_2)$. A subspace $W \subset V$ is a *sub-representation* of (ρ, V) if for every $g \in G$ the matrix $\rho(g)$ maps W to W. A representation is called *irreducible* if it does not have any non-trivial sub-representations. (See Section 3.1.1.2 for formal definitions.)

In this thesis we study the connection between the representations of finite groups and LDCs. We show that if (ρ, V) is an irreducible representation and there exists a small number of elements $g_1, \ldots, g_q$ in G such that some linear combination of $\rho(g_i)$ is a rank one matrix, then we can construct a q-query LDC of length $|G|$ and dimension $\dim V$.[1]

Theorem 1.2.1. *(Informal) Let G be a finite group and let (ρ, V) be an irreducible representation of G with $g_1, \ldots, g_q$ in G and $c_1, \ldots, c_q \in \mathbb{F}$ such that $\mathrm{Rank}(\sum c_i \rho(g_i)) = 1$. Then there exists a $(q, \delta, q\delta)$-locally decodable code $\mathcal{C} : V \to \mathbb{F}^G$.*

This gives a completely new approach to constructing LDCs. Now in order to construct an LDC it is enough to construct irreducible representations with a sparse element of the group algebra of rank one. This theorem gives what we believe is the real algebraic nature behind LDCs.

Given this, we ask a natural question: When can one construct such a representation? We show that in this framework we can achieve the parameters of the best known construction. These construction leads to the same code as from MVCs.

On the connection between Matching Vectors and our approach: The question of when do MV codes fall in the framework of irreducible representations is completely not obvious. We say that MV are *symmetric* if they are an orbit of some group acting on $\mathbb{Z}_m^h$. We show in Section 3.3 that MV Codes can be explained in the framework of irreducible representations for MV that are symmetric. Next we show that the construction given in [23] is symmetric.[2] This gives a way to interpret the construction in [Efr09] as a construction of an irreducible representation. The relationship between MV and LDCs is summarized in the following diagram:

$$\begin{array}{ccc} \text{Irreducible Representation} & \Rightarrow & \text{LDC} \\ \Uparrow & & \Uparrow \\ \text{Symmetric MV} & \Rightarrow & \text{MV} \end{array}$$

We want to mention that all MV codes can be constructed from very specific class of representations of a very specific class of groups. We do not see any reason why such representations will give best possible LDCs although today we do not know how to construct representations which will lead to better codes

Modular Representations and Reed-Muller Codes: One might wonder if the requirement on the representation being irreducible is essential. Perhaps better locally decodable codes can be constructed with reducible representations? We deal with this question in Section 3.4. We distinguish two cases. The first is when the characteristic of the field does not divide the size of the group. In this case we show that irreducibility is essential for our construction to lead to locally decodable codes. The second case is when the characteristic of the field divides the

[1] In fact we prove a stronger statement. For details see Theorem 3.2.1.

[2] In fact, for sake of simplicity, we show it only for a slight modification of Grolmusz [23] construction. While it is true for the Grolmusz construction as well.

size of the group; this brings us to a slightly less familiar territory of representation theory known as *modular representation theory*. We show that in this case, it *is* possible to construct locally decodable codes based on reducible representations. We still don't know, unfortunately, if this can lead to improvements over the best known constructions of LDCs, although this is definitely a promising direction. We illustrate this case by showing how one can view Reed-Muller codes as a special case of our result for reducible modular representations.

1.3 Noise tolerance of LDCs

The main line of research regarding LDCs seeks to identify the shortest possible code length, in terms of the message length n, while keeping the query complexity, the error-rate and the success probability constant.

The decoding algorithm in all of the aforementioned constructions is *smooth*, i.e., each of its queries is uniformly distributed. The analysis of the decoding algorithm relied on all of the queried symbols being uncorrupted. Using the union bound, one could obtain a decoder with success probability greater than half, only if the error-rate was below $\frac{1}{2q}$.

Another line of research focused on improving the error-tolerance of LDCs. Woodruff [42] showed how to increase the handled error-rate from $\frac{1}{2q}$ to $\frac{1}{q}$ over binary alphabets. Dvir, Gopalan and Yekhanin [10], showed how to handle $\frac{1}{4}$ fraction of errors for the codes of [12]. Ben-Aroya et al. [6] showed the same codes could recover from any error-rate below $\frac{1}{2}$. Gal and Mills [15] obtained exponential lower bounds for 3-query LDCs that can tolerate a high error-rate.

When the error-rate is above half of the code's distance, the information in a corrupted codeword is insufficient to uniquely identify the original (uncorrupted) codeword. Thus, in this case, we have to consider *list-decoding*. A code $\mathcal{C}$ is said to be $(1-\alpha, L)$-list-decodable if for every word, the number of codewords within relative distance $1-\alpha$ from that word is at most L. The notion of list-decoding dates back to works by Elias [14] and Wozencraft [44] in the 50s. Roughly speaking, a code $\mathcal{C}$ is *locally* list-decodable if it is $(1-\alpha, L)$-list-decodable, and given a corrupted word w, an index $k \in [L]$ and a target bit j, the decoder returns the j'th message bit of the k'th codeword that is close to w. As expected, there are some subtleties in the definition. The main issue is guaranteeing that for a fixed k, all answers for inputs (k, j) correspond to the same codeword. More formally, a local list-decoding algorithm generates L machines $\{M_k\}$, such that the machine M_k locally decodes one codeword that is close to w, and the machines $\{M_k\}$ together cover all the codewords that are close to w (for a formal definition, see Section 4.1).

The notion of local list-decoding is central in theoretical computer science. It first implicitly appeared in the celebrated Goldreich-Levin result [20], that can be interpreted as a local list-decoding algorithm for the Hadamard code. Later on, many local list-decoding algorithms were studied, especially for Reed-Muller codes [1, 21, 22, 38], direct product and XOR codes [24, 25, 27] and low-rate random codes [30, 33]. In [6] it was shown how to locally list-decode the subexponential-length codes of [12] with only a constant number of queries.

Our Result. In this thesis we show a *generic*, *simple* transformation that takes a locally decodable code $\mathcal{C}$ that can tolerate a low error-rate, and results in a code $\mathcal{C}'$ that can tolerate a much higher fraction of errors. The construction also works in the list-decoding regime, i.e.,

it can transform any LDC $\mathcal{C}$ to a code $\mathcal{C}'$ which is locally list-decodable from an error-rate of $1-\gamma$, for any $\gamma > 0$. Furthermore, the list-decoder for the new code outputs only a constant number of codewords.

The transformation was suggested previously by Trevisan [39], who used it to construct list-decodable codes. We observe that this transformation, when used with a locally-decodable code, results in a locally list-decodable code. While the observation is trivial, it appears to have been unnoticed previously.

The transformation is based on the following idea. An error correcting code with relative distance α is a function $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ that maps any two different strings, to two strings that differ in at least an α fraction of the coordinates. The decoding algorithm can therefore map any string $\tilde{c}$ with more than $(1-\alpha/2)n$ agreement with a codeword $c = \mathcal{C}(\lambda)$, to the correct message λ. We can view this as an $\alpha/2$ to 0 error reduction: given a codeword with some $\alpha/2$ fraction of errors, one can correctly recover the original message.[3].

Similarly, one can define a related notion of codes that only amplify the error-tolerance, without completely correcting the corrupted word. That is, one can design a code $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$, such that given access to a corrupted word $\tilde{c}$ with γn agreement with some codeword $c = \mathcal{C}(\lambda)$, one can compute a message $\tilde{\lambda}$ with some larger $\beta > \gamma$ agreement with λ. We call such a code an *approximately* locally decodable code. When γ is small, several codewords can be γ-close to $\tilde{c}$ and one has to resort to list-decoding. In this case, the code is called an *approximately* locally list-decodable code. Such codes naturally arise in hardness amplification (see, e.g., [26]). For a formal definition see Section 4.1.

Now, let us return to the problem of finding a good locally list-decodable code. Our approach is to compose a locally decodable code (handling the $\alpha/2$ to 0 error reduction) with an approximately locally list-decodable code (handling the $\frac{1}{2}-\epsilon$ to $\alpha/2$ error reduction, for binary codes). Namely, we first encode a message λ with a locally-decodable code $\mathcal{C}$ and then encode the result with an approximately locally list-decodable code to get the code $\mathcal{C}'$. To see that it works, assume we are given a word with $\frac{1}{2}+\epsilon$ agreement with some codeword of $\mathcal{C}'$. We first apply the approximate local list-decoder and get a list of words, each with $1-\alpha/2$ agreement with some codeword of $\mathcal{C}$. We then (uniquely) locally decode each of these corrupted codewords to get λ_i, the i'th symbol of the message λ.

In fact, the local list-decoders of Reed-Muller codes [1, 21, 22, 38] and the Hadamard code [20], also have this two-step structure, combining an error-reduction step (that does not completely correct the corrupted word) with another unique decoding step. The main difference is that Reed-Muller and Hadamard codes are *locally correctable*, i.e., the first error-reduction step returns a close codeword, instead of close message. Therefore, these two steps can be done implicitly without the use of any general approximate list-decoding mechanism. In our case we present a generic transformation that may work with LDCs that are not known to be locally correctable (e.g., the code of [12]) and we therefore need to compose the code with an approximately locally list-decodable code.

Composing the locally decodable binary codes of [12, 35] with the binary approximately locally list-decodable codes of [26] we get:

[3]For a treatment of the related notion of worst-case to average-case reduction and its relationship to error correcting codes see for example [26].

Theorem 1.3.1. *For every $r \geq 2$ there exists a binary code of length*

$$\exp(\exp(O(\sqrt[r]{\log n(\log\log n)^{r-1}}))),$$

which is locally list-decodable from an error-rate of $1/2 - \alpha$. The list-decoding algorithm outputs a list of size $O(\frac{1}{\alpha^2})$ and uses at most $O(\frac{r+\log(1/\epsilon)}{\alpha^3} \cdot 2^r)$ queries.

A locally list-decodable code of similar length was given in [6]. However, the list size in the list-decoding algorithm of [6] was $\mathrm{poly}(n)$, while in Theorem 1.3.1 it is constant. The query complexity of the list-decoding algorithm of [6] was also worse than that of Theorem 1.3.1. On the other hand, the result in [6] shows the code of [12] is locally list-decodable, while Theorem 1.3.1 only shows that some other (related) code is locally list-decodable, and does not state anything about the original code of [12].

1.4 Collaborators

Chapter 4 is based on paper [7] written in a collaboration with Avi Ben-Aroya and Amnon Ta-Shma. Paper [6] is not included in the thesis and is also written in a collaboration with Avi Ben-Aroya and Amnon Ta-Shma. I would also like to thank Danny Gutfreund who participated and contributed to this work at its early stages.

Chapter 2

Matching Vector Codes

2.1 Definitions and Basic Facts

We will use the following standard mathematical notation:

- $[s] = \{1, \ldots, s\}$;
- $\mathbb{F}_q = GF(q)$ is a finite field of q elements;
- $\mathbb{F}^*$ is a multiplicative group of a field;
- $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$ is the set of integers modulo m;
- $\Delta(\vec{x}, \vec{y})$ denotes the Hamming distance between vectors $\vec{x}, \vec{y} \in \mathbb{F}^n$, i.e., the number of indices where $x_i \neq y_i$.
- $\hat{e}_i \in \mathbb{F}^n$ is the i^{th} unit vector i.e., $\hat{e}_i = (\underbrace{0, \ldots, 0}_{i-1}, 1, 0 \ldots, 0)$.

Definition 2.1.1. A code $C : \mathbb{F}^n \mapsto \mathbb{F}^N$ is said to be (q, δ, ε) locally decodable if there exists a randomized decoding algorithm D^w with an oracle access to the received word w such that the following holds:

1. For every message $\vec{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$ and for every $\vec{w} \in \mathbb{F}^N$ such that $\Delta(C(\vec{x}), \vec{w}) \leq \delta N$ it holds that $\Pr(D^w(i) = x_i) \geq 1 - \varepsilon$; i.e., the decoding algorithm is able to recover the relevant symbol even if up to δ fraction of the codeword symbols are corrupted.
2. The algorithm $D^w(i)$ makes at most q queries to w.

A code C is called linear if C is a linear transformation over $\mathbb{F}$. A locally decodable code is called nonadaptive if D makes all its queries simultaneously. Our constructions of locally decodable codes are linear and nonadaptive.

Definition 2.1.2. A code C is said to have a *perfectly smooth decoder* if $D^{C(\vec{x})}(i) = x_i$ for every $\vec{x}$, and each query of $D(i)$ is uniformly distributed over $[N]$.

Fact 2.1.3 (from [40])**.** *Any code with a perfectly smooth decoder which makes q queries is also $(q, \delta, q\delta)$ locally decodable.*

We want to mention that these codes are interesting when $\delta < \frac{|\mathbb{F}|-1}{q|\mathbb{F}|}$. Therefore if we want to handle constant fraction of noise we can use this theorem only when q is constant.

Proof. Note that if the decoding algorithm D queries w in non-corrupted places then D outputs the correct answer. The probability that any specific query will be corrupted is at most δ. By union bound, the probability that some query will be corrupted is at most $q\delta$. Therefore, the decoder outputs the correct answer with the probability of at least $1 - q\delta$. □

We will also use the following fact:

Fact 2.1.4. *For every m co-prime to p there exists a finite field $\mathbb{F} = GF(p^t)$, where $t \leq m$, and an element $\gamma \in \mathbb{F}$ that is a generator of the multiplicative group of size m, i.e., $\gamma^m = 1$ and $\gamma^i \neq 1$ for $i = 1, 2, \ldots, m-1$.*

Proof. Since m is co-prime to p, we have that $p \in \mathbb{Z}_m^*$. Therefore, there exists $t < m$ such that $p^t \equiv 1 \mod m$. Let us set $\mathbb{F} = GF(p^t)$. The size of the multiplicative group $\mathbb{F}^*$ is $p^t - 1$ and therefore it is divisible by m. Let g be a generator of $\mathbb{F}^*$. Then $\gamma = g^{\frac{p^t-1}{m}}$ is a generator of the multiplicative group of size m. □

2.2 Locally Decodable Codes

In our construction we follow Yekhanin's general framework. The construction consists of two parts: The first part is a construction of matching sets of vectors that correspond to "combinatorially nice" sets used in [45]. The second part is a construction of an S-decoding polynomial with a small number of monomials, which correspond to "algebraically nice" sets used in [45]. Let us fix some composite number m for our construction. We will describe a general scheme for the construction of LDCs followed by a concrete example of a 3-query LDC.

2.2.1 Matching Sets of Vectors

Definition 2.2.1. Let x, y be any two elements in $(\mathbb{Z}_m)^h$, where $x = (x_1, x_2 \ldots, x_h), y = (y_1, y_2, \ldots, y_h)$ with $x_i, y_i \in \mathbb{Z}_m$. The bracket product of x, y is: $\langle x, y \rangle = \sum_{i=1}^{h} x_i y_i$.

Definition 2.2.2. For any set $S \subset \mathbb{Z}_m \backslash \{0\}$ a family of vectors $\{u_i\}_{i=1}^n \subseteq (\mathbb{Z}_m)^h$ is said to be *S-matching* if the following conditions hold:

1. $\langle u_i, u_i \rangle = 0$ for every $i \in [n]$.
2. $\langle u_i, u_j \rangle \in S$ for every $i \neq j$.

The goal of this subsection is to construct a large S-matching family over a small domain $(\mathbb{Z}_m)^h$. The main advantage of working with composite numbers comes from the following lemma:

Lemma 2.2.3 (Theorems 1.2 and 1.4 from [23])**.** *Let $m = p_1 p_2 \cdots p_r$ be a product of r distinct primes p_i. Then there exists $c = c(m) > 0$, such that for every integer $h > 0$, there exists an explicitly constructible set-system $\mathcal{H}$ over the universe of h elements (i.e $\mathcal{H}$ is a set of subsets of $[h]$) and there is a set $S \subset \mathbb{Z}_m \setminus \{0\}$ such that:*

1. $|\mathcal{H}| \geq \exp\left(c\frac{(\log h)^r}{(\log\log h)^{r-1}}\right)$;
2. *Size of every set* H *in set-system* $\mathcal{H}$ *is divisible by* m *i.e.,* $|H| \equiv 0 \mod m$;
3. *Let* G, H *be any two different sets in set-system* $\mathcal{H}$*. Then the size of the intersection of* G *and* H *modulo* m *is restricted to be in* S*. That is,* $\forall G, H \in \mathcal{H}$ *such that* $G \neq H$*. It holds that* $|G \cap H| \in S \mod m$;
4. S *is a set of size* $2^r - 1$;
5. $\forall s \in S$ *it holds that* s *(mod* p_i*)* *is* 0 *or* 1 *for all* $i = 1, 2, \ldots, r$.

For our construction we will only need the following simple corollary:

Corollary 2.2.4. *For every* h, r *and integer* $m = p_1 p_2 \ldots p_r$ *there exists a set* S *of size* $2^r - 1$ *and a family of* S*-matching vectors* $\{u_i\}_{i=1}^n \subseteq (\mathbb{Z}_m)^h$ *such that* $n \geq \exp\left(c\frac{(\log h)^r}{(\log\log h)^{r-1}}\right)$.

Proof. Let us take set-system $\mathcal{H}$ as in Lemma 2.2.3. For each set $H \in \mathcal{H}$, we will have one vector $u_H \in (\mathbb{Z}_m)^h$ which is the indicator vector of H. Then it holds that $\langle u_H, u_H \rangle = |H| \equiv 0 \mod m$ and $\langle u_H, u_G \rangle = |H \cap G| \in S \mod m$. □

In Section 2.3 we will develop a simple construction of an S-matching set which is slightly weaker than the construction of [23].

2.2.2 S-Decoding Polynomials

Let us fix some positive integer number m. For our construction we will need a γ which generates a multiplicative subgroup of size m and let us take a field $\mathbb{F}$ which contains such a γ. Recall from Fact 2.1.4 that if m is odd then we can take $\mathbb{F} = GF(2^t)$ for some t. We will first construct a linear code over the field $\mathbb{F}$. In the next section we will show how to reduce the alphabet size.
We will need the following definition:

Definition 2.2.5. For $S \subset \mathbb{Z}_m \backslash \{0\}$, we call a polynomial $P \in \mathbb{F}[x]$ an S-decoding polynomial if the following conditions hold:

- $\forall s \in S: \ P(\gamma^s) = 0$,
- $P(\gamma^0) = P(1) = 1$.

Claim 2.2.6. *For any* S *such that* $0 \notin S$ *there exists an* S*-decoding polynomial* P *with at most* $|S| + 1$ *monomials.*

Proof. Let us take $\tilde{P} = \prod_{s \in S}(x - \gamma^s)$. Then $P(x) = \tilde{P}(x)/\tilde{P}(1)$ is an S decoding polynomial. The degree of P is $|S|$. Thus P has at most $|S| + 1$ monomials. □

2.2.3 Matching Vector Codes

Now we are ready to present the construction of our locally decodable codes.
In order to construct our code we will need S-matching vectors $\{u_i\}_{i=1}^n and\ u_i \in (\mathbb{Z}_m)^h$ and in order to construct a local decoder we will need an S-decoding polynomial P.

Definition 2.2.7 (Matching Vector Codes)**.** The parameters of the Matching Vector Code C are:

- S-matching vectors $\{u_i\}_{i=1}^n,\ u_i \in (\mathbb{Z}_m)^h$
- $\gamma \in \mathbb{F}^*$ is a generator of a multiplicative group of size m.

A linear code $C : \mathbb{F}^n \mapsto \mathbb{F}^{m^h}$ is defined by:

$$C(\hat{e}_i)[x] \triangleq (\gamma^{<u_i,x>})_{x\in(Z_m)^h},$$

where we think of a codeword as a function from $(\mathbb{Z}_m)^h$ to $\mathbb{F}$. By linearity:

$$C(c_1, c_2, \ldots, c_n)[x] \triangleq \sum_{i=1}^{n} c_i \gamma^{\langle u_i,x\rangle}.$$

We will now describe how to retrieve the i'th coordinate of the message.

Since P is an S-decoding polynomial and $\{u_i\}$ are S-matching vectors, $\langle u_j, u_i\rangle \in S$ for $i \neq j$, and therefore it follows that $P(\gamma^{<u_i,u_i>}) = 1$ and $P(\gamma^{<u_j,u_i>}) = 0$ for all $i, j \in [n], i \neq j$. Let $P(x) = a_0 + a_1x^{b_1} + a_2x^{b_2} + \cdots + a_{q-1}x^{b_{q-1}}$.
Let us now define the decoding algorithm $D^w(i)$, where w is a received word with up to δ fraction corrupted coordinates and i is the required coordinate.

Input: Oracle access to the received word w and i index of the symbol to decode.

- Choose $v \in (\mathbb{Z}_m)^h$ at random.
- Query $w(v), w(v + b_1u_i), \ldots, w(v + b_{q-1}u_i)$.
- Output

$$c_i = \gamma^{-<u_i,v>} \left(a_0w(v) + a_1w(v + b_1u_i) + \cdots + a_{q-1}w(v + b_{q-1}u_i)\right). \qquad (2.2.1)$$

Algorithm 1: The Decoding Algorithm

Lemma 2.2.8. *The decoding algorithm D is a Perfectly Smooth Decoder.*

Proof. The algorithm D chooses v uniformly at random. Each of the queries $v, v + b_1u_i, \ldots, v + b_{q-1}u_i$ is uniformly distributed. Therefore, in order to prove that D is a Perfectly Smooth Decoder it is enough to prove that $D^{(C(x))}(i) = x_i$. Note that D^w is a linear mapping so it is enough to prove that $D^{C(e_i)}(i) = 1$ and $D^{C(e_i)}(j) = 0$ for $j \neq i$.

$$D^{C(e_i)}(i) = (\gamma^{-<u_i,v>})(a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v+b_1u_i>} + \cdots + a_{q-1}\gamma^{<u_i,v+b_{q-1}u_i>}).$$

But $\langle u_i, v + cu_i \rangle = \langle u_i, v \rangle + c\langle u_i, u_i \rangle = \langle u_i, v \rangle$. So we have,

$$D^{C(e_i)}(i) = \gamma^{-<u_i,v>}(a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v>} + \cdots + a_{q-1}\gamma^{<u_i,v>}) = \\ = a_0 + a_1 + \cdots + a_{q-1} = P(1) = 1.$$

Now let us prove that

$$\forall i \neq j \;\; D^{C(e_i)}(j) = 0.$$

We need to show that

$$a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v+b_1u_j>} + \cdots + a_{q-1}\gamma^{<u_i,v+b_{q-1}u_j>} = 0.$$

Recall that $P(\gamma^{<u_i,u_j>}) = 0$. Therefore,

$$\gamma^{<u_i,v>}(a_0 + a_1\gamma^{b_1<u_i,u_j>} + \cdots + a_{q-1}\gamma^{b_{q-1}<u_i,u_j>}) = \gamma^{<u_i,v>}P(\gamma^{<u_i,u_j>}) = 0.$$

□

The dimension of the code we have constructed is n which is the number of S-matching vectors. The codeword length is $\left|(\mathbb{Z}_m)^h\right| = m^h$ and the number of queries is equal to the number of monomials of P. Therefore, an immediate corollary of Lemma 2.2.8 is:

Theorem 2.2.9. *For any S-matching vectors $\{u_i\}_{i=1}^n \subseteq (\mathbb{Z}_m)^h$ and any S-decoding polynomial with q monomials there exists a $(q, \delta, q\delta)$ locally decodable code $C : \mathbb{F}^n \mapsto \mathbb{F}^{m^h}$.*

An immediate corollary from Corollary 2.2.4 and Claim 2.2.6 is that we can choose $n \geq \exp(c\frac{(\log h)^r}{(\log\log h)^{r-1}})$ and an S-decoding polynomial with less than 2^r monomials. Thus, we have the following theorem:

Theorem 2.2.10. *For any possitive integer $r > 0$ and for every n there exists a $(q, \delta, q\delta)$ locally decodable code $C : \mathbb{F}^n \mapsto \mathbb{F}^N$ with codeword length*

$$N = \exp(\exp(c(r)(\sqrt[r]{\log n(\log\log n)^{r-1}}))),$$

where $c(r)$ is a constant that depends only on r and with a number of queries $q \leq 2^r$.

Proof. Let $m = p_1 \dots p_r$ be the product of r primes. Fix $h = \exp\left(\left(O(\sqrt[r]{\log n(\log\log n)^{r-1}})\right)\right)$. From Corollary 2.2.4 there exists a set S of size $2^r - 1$ and $n = \exp(c\frac{(\log h)^r}{(\log\log h)^{r-1}})$ S-matching vectors. Using the construction above we get a code C with codeword length m^h and a message length n. Fix m to be a constant. Then $m^h = \exp(O(h))$. Therefore,

$$m^h = \exp(O(h)) = \exp\left(\exp\left(O\left(\sqrt[r]{\log n(\log\log n)^{r-1}}\right)\right)\right).$$

From Claim 2.2.6 there exists an S-decoding polynomial with $q \leq 2^r$ monomials. Using this polynomial for our decoding algorithm we get from Lemma 2.2.8 that C has a Perfectly Smooth Decoder which makes q queries. Thus, from Fact 2.1.3 we have that the code C is a $(q, \delta, q\delta)$-LDC. □

The Claim 2.2.6 gives a trivial polynomial with 2^r monomials. This allows us to construct LDCs with 4 queries by setting $r = 2$. In order to construct 3 query LDCs we need to find a polynomial with 3 monomials. Let us give a concrete example of an S-decoding polynomial with 3 monomials. We found this example by an exhaustive search.

Example 2.2.11. *Let $m = 511 = 7 \cdot 73$ and let $S = \{1, 365, 147\}$. By Corollary 2.2.4 there exists S-matching vectors $\{u_i\}_{i=1}^n$, $u_i \in (\mathbb{Z}_m)^h$, where $n \geq \exp(c\frac{(\log h)^2}{\log\log h})$. Set*

$$\mathbb{F} = GF(2^9) = \mathbb{F}_2[\gamma]/(\gamma^9 + \gamma^4 + 1).$$

It can be verified that γ is a generator of $\mathbb{F}^*$ and that the polynomial $P(x) := \gamma^{423} \cdot x^{65} + \gamma^{257} \cdot x^{12} + \gamma^{342}$ is an S decoding polynomial with 3 monomials.

An immediate corollary from this example and Theorem 2.2.10 is 3-query LDC.

Theorem 2.2.12. *There exists a $(3, \delta, 3\delta)$ locally decodable code $C : \mathbb{F}^n \mapsto \mathbb{F}^N$ with $N = \exp(\exp(O(\sqrt{\log n \log\log n})))$.*

We want to mention that in the subsequent work [28] Itoh and Suzuki showed a general way to improve query complexity in Theorem 2.2.10 using sparse S decoding polynomials. They showed how to improve query complexity from 2^r to $3 \cdot 2^{r-2}$ using a single example of S decoding polynomial with 3 monomials. Later Chee at al in [35] showed how to construct infinitely many such examples based on a number theoretic conjecture. Together with [28] in this thesis we improves query complexity to $3^{\lceil r/2 \rceil}$.

2.3 A Simple Construction of S-Matching Vectors

First for our construction we need to define tensor product.

Definition 2.3.1 (Tensor Product)**.** Let R be a ring and let $\vec{x}, \vec{y} \in R^n$. The *tensor product of* $\vec{x}, \vec{y}$ denoted by $\vec{x} \otimes \vec{y} \in R^{n^2}$, is defined by $\vec{x} \otimes \vec{y}(i, j) \triangleq x_i \cdot y_j$, (where we identify $[n^2]$ with $[n] \otimes [n]$.) In the same way we define *the ℓ'th tensor power* $\vec{x}^{\otimes \ell} \in R^{n^\ell}$ by

$$\vec{x}^{\otimes \ell}(i_1, i_2, \ldots, i_\ell) \triangleq \prod_{j=1}^{\ell} x_{i_j}. \tag{2.3.1}$$

We will use only the following fact about tensor products:

Fact 2.3.2.

$$\langle u^{\otimes \ell}, v^{\otimes \ell} \rangle = \langle u, v \rangle^\ell$$

Proof.

$$\begin{aligned} \langle u^{\otimes \ell}, v^{\otimes \ell} \rangle = \sum_{1 \leq i_1, i_2, \ldots, i_l \leq m} \left(\prod_{j=1}^{\ell} u_{i_j} \prod_{j=1}^{\ell} v_{i_j} \right) = \\ \left(\sum_{1 \leq i_1 \leq m} u_{i_1} v_{i_1} \right) \ldots \left(\sum_{1 \leq i_\ell \leq m} u_{i_\ell} v_{i_\ell} \right) = \langle u, v \rangle^\ell. \end{aligned} \tag{2.3.2}$$

□

Lemma 2.3.3. *Let $p_1 < p_2 < \ldots < p_r$ be any r primes and $m = p_1 \cdot p_2 \cdots p_r$. Then for every t, there exists a set S of size $2^r - 1$ and a family of S-matching vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ such that $n = \binom{t}{m-1}$ and $h = O(t^{p_r - 1})$.*

Proof. Let us first construct a family of vectors $\{u'_i\}_{i=1}^n, u'_i \in (\mathbb{Z}_m)^{t+1}$ such that:

1. $\langle u'_i, u'_i \rangle = 0$ for $i \in [n]$.
2. $\langle u'_i, u'_j \rangle \neq 0$ for $i \neq j$.

Identify the subsets of $[t] = \{1, 2, \ldots, t\}$ of size $m-1$ with $\{1, \ldots, \binom{t}{m-1}\}$. For every subset $A \subseteq [t]$ of size $m-1$, let $u'_i \in \mathbb{Z}_m^t$ be the indicator vector of the set, i.e., $u'_i = (a_1, a_2, \ldots, a_t)$, where $a_i = 1$ if $i \in A$ and $a_i = 0$ otherwise. In order to simplify the construction let us add an additional coordinate which is always one i.e., $u'_i = (a_1, a_2, \ldots, a_t, 1)$. Clearly $\langle u'_i, u'_i \rangle = 0$ since u'_i has exactly m ones and $\langle u'_i, u'_j \rangle = 1 + |A_i \cap A_j| \neq 0$. Since intersection of two different subsets of size $m-1$ is always less than $m-1$.

Now we want to change these vectors such that the inner product of two such vectors will be in some small set S. By the Chinese reminder theorem $\mathbb{Z}_m \approx \mathbb{Z}_{p_1} \oplus \mathbb{Z}_{p_2} \oplus \cdots \oplus \mathbb{Z}_{p_r}$. Therefore, any number x in $\mathbb{Z}_m$ we can view as $(x \mod p_1, x \mod p_2, \ldots, x \mod p_r)$. The set S is the set $\{0,1\}^r \backslash (0, 0, \ldots, 0)$ i.e. $a \in S$ iff $a \neq 0$ and for every $k = 1, \ldots, r$ holds $(a \mod p_k) \in \{0, 1\}$.

By the Chinese reminder theorem there exist constants $c_1, c_2, \ldots, c_r \in \mathbb{Z}_m$ such that:

1. $c_i \equiv 1 \mod p_i$
2. $c_i \equiv 0 \mod p_j$ for $i \neq j$

Let us define u_i by:

$$u_i = (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots, c_r u_r'^{\otimes p_r - 1}).$$

Now we need to prove that $\langle u_i, u_i \rangle \equiv 0$:

$$\langle u_i, u_i \rangle = \langle (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots, c_r u_r'^{\otimes p_r - 1}), (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots, c_r u_r'^{\otimes p_r - 1}) \rangle = \sum_{j=1}^r c_j^2 \langle u_i'^{\otimes p_j - 1}, u_i'^{\otimes p_j - 1} \rangle = \sum_{j=1}^r c_j^2 \langle u'_i, u'_i \rangle^{p_j - 1},$$

where the last equation follows from Fact 2.3.2. Since $\langle u'_i, u'_i \rangle = 0$ it follows that $\langle u_i, u_i \rangle = 0$. Now let us prove that $\langle u_i, u_j \rangle \in S$ for any $i \neq j$. In order to prove that $\langle u_i, u_j \rangle \in S$ we will prove that $\langle u_i, u_j \rangle \mod p_k \in \{0, 1\}$ and $\langle u_i, u_j \rangle \neq 0$. Observe that

$$u_i \mod p_k \equiv (0, 0, \ldots, u_i'^{\otimes (p_k - 1)}, 0, \ldots, 0).$$

Thus it follows that:

$$\langle u_i, u_j \rangle \mod p_k \equiv \langle u_i'^{\otimes p_k - 1}, u_j'^{\otimes p_k - 1} \rangle \equiv \langle u'_i, u'_j \rangle^{p_k - 1}$$

By Fermat's Little Theorem $x^{p_k - 1} \equiv 0$ or $1 \mod p_k$ for every k. Since $\langle u'_i, u'_j \rangle \neq 0 \mod m$ for some k we have $\langle u'_i, u'_j \rangle \neq 0 \mod p_k$. Therefore $\langle u_i, u_j \rangle = \langle u'_i, u'_j \rangle^{p_k - 1} \neq 0 \mod p_k$. Therefore $\langle u_i, u_j \rangle \neq 0 \mod m$. □

As a corollary we get:

Corollary 2.3.4. *For every h, r there exists an integer $m = p_1 p_2 \dots p_r$ and a set $S \subset \mathbb{Z}_m \backslash \{0\}$ of size $2^r - 1$ and a family of S-matching vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ such that $n \geq \exp(c\frac{(\log h)^r}{(\log\log h)^{r-1}})$.*

Note that the only difference between Corollary 2.3.4 and Corollary 2.2.4 is in order of quantifiers i.e. Corollary 2.2.4 holds for every m while Corollary 2.3.4 holds for some specific m.

Proof of Corollary 2.3.4. Let us take all primes of the same size (i.e. $p_i = p_j + o(p_i)$) and $t = m^2$; then in Lemma 2.3.3 we will get that $n \geq \binom{m^2}{m-1} \geq m^m = O(m^{p^r})$ and $h = O(m^{2p_r})$. Thus it follows that:

$$n \geq \exp(c\frac{(\log h)^r}{(\log\log h)^{r-1}}).$$

□

2.4 Binary Locally Decodable Codes

In this section we will show how to reduce the alphabet size from p^t to p. The cost of the reduction will be factor q (q is the number of queries) in length of the code and factor $\frac{p}{p-1}$ in smoothness. It will not depend on t. Thus, by taking $p = 2$ we will get binary codes.

For our reduction we need the following simple lemma:

Lemma 2.4.1. *For every m, h there exists a linear functional $L : \mathbb{F}_{p^t} \mapsto \mathbb{F}_p$ such that*

$$\forall i \in [n] \Pr_{v \in (\mathbb{Z}_m)^h}(L(\gamma^{<u_i, v>}) \neq 0) \geq 1 - \frac{1}{p}.$$

Proof. Observe that for random v, $\langle u_i, v\rangle$ is a random number in $\mathbb{Z}_m$, since the GCD of u_i's coordinates is 1. Thus it is enough to find L such that

$$\Pr_{j \in \mathbb{Z}_m}(L(\gamma^j) \neq 0) \geq 1 - \frac{1}{p}.$$

For a constant j and a random L, $\Pr(L(\gamma^j) \neq 0) = 1 - \frac{1}{p}$ thus, the expectation of $\Pr_{j \in \mathbb{Z}_m}(L(\gamma^j) \neq 0)$ is $1 - \frac{1}{p}$ i.e.,

$$\mathbf{E}_L(\Pr_{j \in \mathbb{Z}_m}(L(\gamma^j) \neq 0)) = 1 - \frac{1}{p}.$$

Therefore, there exists an L such that

$$\Pr_{j \in \mathbb{Z}_m}(L(\gamma^j) \neq 0) \geq 1 - \frac{1}{p}.$$

□

Let us describe the reduction formally: Choose L such that $\Pr_{j \in \mathbb{Z}_m}(L(\gamma^j) \neq 0) \geq 1 - \frac{1}{p}$. Since m is a constant we can find it by an exhaustive search in constant time.

1. Given a message $(c_1, c_2, \ldots, c_n)$, encode it with the code from the previous section $w = C(c_1, c_2, \ldots, c_n)$.

2. Extend it to

$$\tilde{w} \triangleq \tilde{w_0} \circ \tilde{w_1} \circ \cdots \circ \tilde{w}_{q-1} \triangleq a_0 w \circ a_1 w \circ \cdots \circ a_{q-1} w.$$

3. Reduce the alphabet by applying L on every symbol of $\tilde{w}$ and return

$$w_0 \circ w_1 \circ \cdots \circ w_{q-1} \triangleq L(\tilde{w_0}) \circ L(\tilde{w_1}) \circ \cdots \circ L(\tilde{w}_{q-1}).$$

Let us define the decoding algorithm $D^w(i)$:

Input: Oracle access to the received word w and i index of the symbol to decode.

- Choose $v \in (\mathbb{Z}_m)^h$ at random conditioned on $L(\gamma^{<u_i,v>}) \neq 0$.
- Query $w_0(v), w_1(v + b_1 u_i), \ldots, w_{q-1}(v + b_{q-1} u_i)$.
- Output $c_i = L(\gamma^{<u_i,v>})^{-1}(w_0(v) + w_1(v + b_1 u_i) + \cdots + w_{q-1}(v + b_{q-1} u_i))$.

Algorithm 2: The Decoding Algorithm

Theorem 2.4.2. *The binary code C defined above is $(q, \delta, \frac{p}{p-1} q\delta)$ locally decodable.*

Proof. We will prove it in two steps: First let us prove that if at most δ fraction of the codeword $w = w_0 \circ w_1 \circ \cdots \circ w_{q-1}$ is corrupted then we query a corrupted place with probability at most $\frac{p}{p-1} q\delta$. Let δ_i be a fraction of corrupted bits in w_i so $\frac{1}{q} \sum \delta_i = \delta$. We chose L such that v is distributed uniformly among $\frac{p-1}{p}$ fraction of all possible values. Therefore, the probability that query i will be corrupted is at most $\frac{p}{p-1}\delta_i$. So the probability that one of the queries will be corrupted is at most $\sum \frac{p}{p-1}\delta_i = \frac{p}{p-1} q\delta$.

Next let us prove that if we query only non-corrupted places then we will return the correct answer. As before, by linearity it is enough to prove that $D^{C(e_i)}(i) = 1$ and $D^{C(e_j)}(i) = 0$ for $i \neq j$.

$$\begin{aligned} D^{C(e_i)}(i) &= L(\gamma^{<u_i,v>})^{-1}(L(a_0\gamma^{<u_i,v>}) + L(a_1\gamma^{<u_i,v+b_1u_i>}) + \cdots + L(a_{q-1}\gamma^{<u_i,v+b_{q-1}u_i>})) \\ &= L(\gamma^{<u_i,v>})^{-1} L\left(\textstyle\sum_{j=0}^{q-1} a_j \gamma^{<u_i,v+b_j u_i>}\right) = L\left(\textstyle\sum_{j=0}^{q-1} a_j \gamma^{<u_i,v>}\right) \\ &= L(\gamma^{<u_i,v>})^{-1} L(P(1)\gamma^{<u_i,v>}) = L(\gamma^{<u_i,v>})^{-1} L(\gamma^{<u_i,v>}) = 1. \end{aligned}$$

In the same way we can prove that $D^{C(e_j)}(i) = 0$.

$$\begin{aligned} D^{C(e_j)}(i) &= L(\gamma^{<u_i,v>})^{-1}(L(a_0\gamma^{<u_j,v>}) + L(a_1\gamma^{<u_j,v+b_1u_i>}) + \cdots + L(a_{q-1}\gamma^{<u_j,v+b_{q-1}u_i>})) \\ &= L(\gamma^{<u_i,v>})^{-1} L\left(\gamma^{<u_j,v>} \textstyle\sum_{t=0}^{q-1} a_t \gamma^{b_t <u_j,u_i>}\right) = L(\gamma^{<u_i,v>})^{-1} L\left(P(\gamma^{<u_i,u_j>})\gamma^{<u_i,v>}\right) \\ &= L(\gamma^{<u_i,v>})^{-1} L(0) = 0. \end{aligned}$$

□

2.5 Future Work

In this chapter we give a general construction of LDCs from any S-matching set and any S-decoding polynomial. Any improvement in size of a set-system with restricted intersections will immediately yield an improvement in the rate of LDCs. We hope that this thesis will give motivation for future work on set-systems with restricted intersections.

Chapter 3

Locally Decodable Codes from Irreducible Representations

3.1 Notation and Preliminaries

3.1.1 Representation Theory

In this section we give basic facts about representation theory. We do not give proofs here and the interested reader is referred to any standard textbook on the subject such as [36].

3.1.1.1 Group Action

First let us start with the definition of the action of a group on a set.

Definition 3.1.1. *We say that a group G acts on a set X if there exists a mapping $T : G \times X \to X$ such that $T(g_2, T(g_1, x)) = T(g_2 g_1, x)$ and $T(1, x) = x$.*

Usually the action is obvious from the context and then we write $g \cdot x$ instead of $T(g, x)$. Note that each $g \in G$ defines a permutation on the set X.

Definition 3.1.2. *We say that G acts transitively on the set X iff for every $x, y \in X$ there exists $g \in G$ such that $gx = y$. In this case we say that X is an orbit of G.*

Let us assume that G acts on the set X. Then using this action we can define a new action of the group G on Σ^X. It is more convenient to view Σ^X as the set of functions from X to Σ rather than a string of symbols, i.e., we view $f \in \Sigma^X$ as $f : X \to \Sigma$.

Definition 3.1.3. *Suppose G acts on the set X. Define an action of G on Σ^X by $(gf)(x) = f(g^{-1}x)$. We call such an action a permutation action.*

Note that we need to prove that this is indeed an action. That is, we need to check that $(g_1 \cdot (g_2 \cdot f)) = (g_1 \cdot g_2) \cdot f$. Note also that if we view Σ^X as a set of strings, then G acts on it by permuting coordinates.

Definition 3.1.4. *An* order *of the group G is a minimal number m such that for every $g \in G$ it holds that $g^m = 1$.*

Definition 3.1.5. *We say that the group H* acts on the group *N if it acts on it as a set and for every $h \in H, n_1, n_2 \in N$ it holds that*

$$h \cdot (n_1 n_2) = (h \cdot n_1)(h \cdot n_2) \,. \tag{3.1.1}$$

Note that action on the group in particular is an action on the set but the converse is not true. Any group N has a natural action on itself as a set. Note that this action does not satisfy Equation 3.1.1. Therefore N acts on itself as a set by not as a group.

Definition 3.1.6 (Semi-Direct Product of Groups)**.** *Let N be a group. Let H be a group acting on the group N. Then the semi-direct product of N by H denoted by $N \rtimes H$ is a sub-group of permutations of N. Generated by the permutations defined by the actions of N on H and the natural actions of N on itself.*

3.1.1.2 Group Representations

Notation 3.1.1. *We denote by $\mathrm{Mat}(V)$ the set of all matrices on the vector space V. $GL(V)$ denotes the group of invertible matrices on the vector space V.*

Definition 3.1.7 (Representation of a Group)**.** *A representation (ρ, V) of a group G in a vector space V is a group homomorphism $\rho : G \to GL(V)$, that is, for every $g_1, g_2 \in G$ it holds that $\rho(g_1) \cdot \rho(g_2) = \rho(g_1 \cdot g_2)$.*

We also can define a representation of group G as an action of G on the vector space as follows:

Definition 3.1.8. *Let V be a vector space over the field $\mathbb{F}$. A representation of a group G in V is an action of the group G on the set V which satisfies the following conditions:*

- *For any $v_1, v_2 \in V$ it holds that $g \cdot (v_1 + v_2) = g \cdot v_1 + g \cdot v_2$.*
- *For any $\lambda \in \mathbb{F}$ it holds that $g \cdot (\lambda v) = \lambda g \cdot v$.*
- *For any $v \in V$ it holds that $1 \cdot v = v$.*

Definition 3.1.9 (Sub-Representation)**.** *Let ρ be a representation of a group G in a vector space V. We say that $U \subset V$ is a sub-representation of ρ if U is a linear subspace of V and U is invariant under ρ, namely: for every $g \in G$ it holds that $\rho(g)U = U$.*

Definition 3.1.10 (Irreducible-Representation)**.** *Let ρ be a representation of a group G in a vector space V. We say that ρ is an irreducible representation if it does not have any non trivial sub-representations, else we say that ρ is reducible.*

We need the following decomposition theorem:

Theorem 3.1.2 (Complete Reducibility)**.** *Let G be a group. Let V be a vector space over an algebraically closed field $\mathbb{F}$ of characteristic co-prime to the size of G. Let ρ be a representation of the G in the vector space V. Then $V = \oplus V_i$ where V_i are irreducible sub-representations of ρ.*

The following theorem says that any orbit of an irreducible representation spans the entire space.

Lemma 3.1.3. Let (ρ, V) be an irreducible representation of G. Let $v \in V$ be a non-zero vector. Then the set $\{\rho(g)v | g \in G\}$ spans V, and thus there exist $g_1, g_2, \ldots, g_k \in G$ such that $\{\rho(g_i)v\}_{i=1}^k$ is a basis for V.

3.1.1.3 Homomorphisms between Representations

Definition 3.1.11. *Let ρ_1 be a representation of the group G in a vector space V and ρ_2 be a representation of the group G in a vector space W. We say that a linear mapping $T : V \to W$ is a homomorphism from (ρ_1, V) to (ρ_2, W) iff $\forall g \in G$ $\rho_2(g) \circ T = T \circ \rho_1(g)$. Sometimes we also say that T is a G-homomorphism.*

Schematically a linear mapping T is a homomorphism between (ρ_1, V) and (ρ_2, W) if the following diagram is commutative:

$$\begin{array}{ccc} V & \xrightarrow{T} & W \\ \rho_1(g) \downarrow & & \downarrow \rho_2(g) \\ V & \xrightarrow{T} & W \end{array}$$

We say that a homomorphism T from (ρ_1, V) to (ρ_2, W) is an embedding/isomorphism if T is an embedding/isomorphism of the vector spaces V and W. Note also that kernel of T is a sub-representation. Thus if (ρ_1, V) is irreducible then T is either embedding or zero homomorphism.

Lemma 3.1.4. Let (ρ, V) be a representation of G. Let $\{V_i\}_{i=1}^k$ be irreducible non-isomorphic sub-representations of V. Then vector spaces $\{V_i\}_{i=1}^k$ are linearly independent.

3.1.1.4 Permutational Representation, Group Algebra

Assume that a group G acts on a set X. Consider the permutation action of G on $\mathbb{F}^X$. It is easy to see that this action admits the properties of Definition 3.1.8. Thus we can define a representation τ of the group G in $\mathbb{F}^X$. We call τ the *permutational representation*. Specifically:

$$(\tau(g) \cdot f)(x) = f(g^{-1}x). \tag{3.1.2}$$

For any $f \in \mathbb{F}^X$ we define support of f by the number of non-zero entries of $f : X \to \mathbb{F}$ i.e.,

$$\text{supp}(f) = |\{x \in X | f(x) \neq 0\}| .$$

For linear subspace $U \subset F^X$, we define support as a union of supports of all vectors in U, i.e.,

$$\text{supp}(U) = |\cup_{f \in U} \{x \in X | f(x) \neq 0\}| .$$

Lemma 3.1.5. Let U be a vector subspace of $\mathbb{F}^X$ of the full support and let $|\mathbb{F}| \geq t$. Then there exist a vector $u \in U$ such that $\text{supp}(u) \geq (1 - \frac{1}{t})|X|$.

Now let us define the group algebra $\mathbb{F}[G]$:

Definition 3.1.12 (Group Algebra)**.** *The group algebra $\mathbb{F}[G]$ is the set of all functions from G to $\mathbb{F}$. Multiplication in this group algebra is given by*

$$(f * h)(x) = \sum_{g_1 \cdot g_2 = x} f(g_1)h(g_2) \ .$$

We write $f \in \mathbb{F}[G]$ as a formal sum: $f = \sum_{i=1}^{n} a_i g_i$ meaning that $f(g_i) = a_i$ for $g_1, g_2, \ldots, g_n$ and zero on the rest of G. We say that $f \in \mathbb{F}[G]$ is a q-sparse element if it has support of size at most q i.e., it can be written in the form $f = \sum_{i=1}^{q} a_i g_i$.

Definition 3.1.13 (Regular Representation)**.** *The regular representation of the group G is the representation ρ in the group algebra $\mathbb{F}[G]$ given by: $\rho(g)f = g * f$.*

Note that an equivalent way to define the regular representation is as a permutational representation of $\mathbb{F}^G$, where the group G acts on G in a natural way.

The regular representation plays an important role since it contains all irreducible representations. It follows from the following basic theorem from representation theory.

Theorem 3.1.6. *Let V be a vector space over the field $\mathbb{F}$. Then for every irreducible representation (ρ, V) there exists some G-embedding from V to $\mathbb{F}[G]$.*

Notation 3.1.7. *Let $\rho : G \to GL(V)$ be any representation of the group G. Then we can linearly extend ρ to the group algebra $\mathbb{F}[G]$ i.e., $\rho : \mathbb{F}[G] \to \mathrm{Mat}(V)$ where $\rho(f)$ is defined as $\sum_{g \in G} f(g)\rho(g)$. Note that now $\rho(f)$ may be any matrix, not necessary invertible.*

Note that if $(\rho_1, V), (\rho_2, W)$ are two representations and $T : V \to W$ is a homomorphism between them, then for any $f \in \mathbb{F}[G]$ it holds that

$$T \circ \rho_1(f) = \rho_2(f) \circ T. \tag{3.1.3}$$

3.1.1.5 Dual Space, Dual Representation

Definition 3.1.14. *Let V be a linear vector space over field $\mathbb{F}$. Then the* dual space *of V, denoted by V^* is the set of all linear functionals from V to $\mathbb{F}$.*

We want to mention here that $\dim V = \dim V^*$.

Definition 3.1.15. *Let V be a vector space of dimension k. Let $u_1, u_2, \ldots, u_k$ be a basis of V and $v_1, \ldots, v_k$ be a basis of V^*. We say these bases are* dual *if $v_i(u_j) = \delta_{i,j}$, where δ_{ij} is Kronecker delta i.e., $\delta_{ij} = 1$ is iff $i = j$ and zero otherwise.*

Theorem 3.1.8. *For every basis there exists a dual basis.*

Now let us define the dual representation:

Definition 3.1.16 (Dual Representation)**.** *Let V be a vector space over $\mathbb{F}$. Let (ρ, V) be a representation of the group G. Let V^* be the set of all linear functionals from V to $\mathbb{F}$. The dual representation $(\bar{\rho}, V^*)$ is given by $\bar{\rho}(g)(\ell) = \ell \circ \rho(g^{-1})$, i.e., $\bar{\rho}(g)(\ell)(v) = \ell(\rho(g^{-1})v)$.*

Note that $\dim V = \dim V^*$. Also it holds that $(V^*)^* = V$. We leave to the reader to check that this is indeed a representation. In many cases a representation is isomorphic to its dual representation, but not always. However, a representation is irreducible if and only if its dual is irreducible.

Theorem 3.1.9. *The representation (ρ, V) is irreducible if and only if $(\bar{\rho}, V^*)$ is irreducible.*

The dual group for Abelian groups is very similar to dual representation.

Definition 3.1.17. *Let A be an Abelian group. Let m be an order of A. A dual group A^* is a set of group homomorphisms $\theta : A \to \mathbb{Z}_m$.*

Note that if θ_1, θ_2 are group homomorphisms then $\theta_1 + \theta_2$ is also group homomorphism. Therefore, A^* is an Abelian group. Moreover A^* isomorphic to A. For example if $A = \mathbb{Z}_m^h$ then isomorphism is given by $a \mapsto \langle -, a \rangle$. If some group H acts on an Abelian group A then it also acts on its dual A^*. For $h \in H$ and $\theta \in A^*$ the action of is given by the rule $h \cdot \theta(x) = \theta(h^{-1} \cdot x)$.

3.1.2 Locally Decodable Codes

Definition 3.1.18. *A code $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^n$ is said to be (q, δ, ε) locally decodable if there exists a randomized decoding algorithm D^w with an oracle access to the received word w such that the following holds:*

1. *For every message $m = (m_1, m_2, \ldots, m_k) \in \mathbb{F}^k$ and for every $w \in \mathbb{F}^n$ such that $\Delta(\mathcal{C}(m), w) \leq \delta n$ for every i, it holds that $\Pr(D^w(i) = m_i) \geq 1 - \varepsilon$, where probability is taken over internal randomness of D. This means that the decoding algorithm can recover the relevant symbol even if up to δ fraction of the codeword symbols are corrupted.*
2. *The algorithm $D^w(i)$ makes at most q queries to w.*

A code $\mathcal{C}$ is called linear if $\mathcal{C}$ is a linear transformation over $\mathbb{F}$. A locally decodable code is called non-adaptive if D makes all its queries simultaneously. Our constructions of locally decodable codes are linear and non-adaptive.

Definition 3.1.19. *A code $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^n$ is said to have a* c-smooth decoder *if $D^{\mathcal{C}(m)}(i) = m_i$ for every $m \in \mathbb{F}^k$ and for every i. Each query of $D(i)$ is uniformly distributed over a domain of size cn.*

3.2 Locally Decodable Codes from Irreducible Representations

Let us start from the main theorem of this chapter.

Theorem 3.2.1. *Let G be a group acting on a set X. Let $(\tau, \mathbb{F}^X)$ be the permutational representation defined by this action. Let (ρ, V) be a representation of G. Let $\mathcal{C} : V \to \mathbb{F}^X$ be a G-homomorphism between representations (ρ, V) and $(\tau, \mathbb{F}^X)$. Assume that the following conditions hold:*

1. *(a) There exists a q-sparse element $D \in \mathbb{F}[G]$, $D = \sum_{i=1}^{q} c_i g_i$ such that $\mathrm{Rank}(\rho(D)) = 1$.*
   (b) (ρ, V) is an irreducible representation.
2. *Let $v \in \mathrm{Im}(\rho(D))$ be a non-zero vector.*[1] *Then supp$(\mathcal{C}(v)) \geq c|X|$.*

[1] Note that since $\mathrm{Rank}(\rho(D)) = 1$, the vector v is unique up to scalar multiplication.

Let $k = \dim V$. Then there exists a basis $b_1, \ldots, b_k$ for V such that

$$(m_1, m_2, \ldots, m_k) \mapsto \mathcal{C}(\sum_{i=1}^{k}(m_i b_i))$$

is a $(q, \delta, \frac{q\delta}{c})$-Locally Decodable Code.

We want to mention that this theorem is non-trivial for $\delta < \frac{c}{q}$ thus if we want to construct codes which can handle constant fraction of noise we need that c in the theorem will be constant.

In Subsection 3.2.2 we show that if one constructs a representation ρ that satisfies Condition 1 of Theorem 3.2.1, then we can always embed it into the regular representation in a way that satisfies Condition 2 of the theorem. In Subsection 3.2.3 we show that if $\mathbb{F}$ is an algebraic extension of $\mathbb{F}_p$, we can reduce the alphabet to $\mathbb{F}_p$ almost at no cost. In Section 3.4 we show that when $|\mathbb{F}|$ and $|G|$ are co-prime then the irreducibility of (ρ, V) is essential for having a rank one element. Moreover, we show that (ρ, V) should be irreducible not only over $\mathbb{F}$ but also over the algebraic closure of $\mathbb{F}$.

Proof. The proof is divided into two parts. The first part is Lemma 3.2.2 which constructs a basis for V. This basis defines the encoding algorithm. In the second part we construct a decoding algorithm with q queries and show that it is a c-smooth decoder.

Lemma 3.2.2. There exists a basis $\{b_1, b_2, \ldots, b_k\}$ for V and $h_1, \ldots, h_k \in G$ such that $b_i \in \mathrm{Ker}(\rho(D * h_j))$ if and only if $i \neq j$.

Proof. Set $L = \mathrm{Ker}\,\rho(D)$. L is a linear subspace of V of dimension $k-1$. Therefore, there exists unique (up to scalar multiplication) non-zero linear functional $u \in V^*$ such that $u(L) = 0$. Since (ρ, V) is an irreducible representation, it follows by Theorem 3.1.9 that its dual $(\bar{\rho}, V^*)$ is also irreducible. Therefore, from Lemma 3.1.3[2] it follows that there exist $h_1^{-1}, h_2^{-1}, \ldots, h_k^{-1} \in G$ such that $\{\bar{\rho}(h_i^{-1})u\}_{i=1}^k$ is a basis for V^*. By Theorem 3.1.8 it follows that for this basis there exists a dual basis $\{b_1, b_2, \ldots, b_k\}$ for V. From the definition of the dual basis it holds that $(\bar{\rho}(h_i^{-1})u)(b_j) = \delta_{ij}$. Thus $b_i \in \mathrm{Ker}\,\bar{\rho}(h_i^{-1})u$ if and only if $i \neq j$. In order to complete the proof of the lemma we need to show that $\mathrm{Ker}(\bar{\rho}(h_i^{-1})u) = \mathrm{Ker}\,\rho(D * h_i)$. Let $v \in \mathrm{Ker}\,\rho(D * h_i)$ then $0 = \rho(D * h_i)v = \rho(D)\rho(h_i)v$. Thus $\rho(h_i)v \in \mathrm{Ker}\,\rho(D)$ by definition of u it also holds that $u(\rho(h_i)v) = 0$. Therefore $\bar{\rho}(h_i^{-1})u(v) = 0$. □

Let $b_1, \ldots, b_k$ and $h_1, \ldots, h_k$ be given by Lemma 3.2.2. The encoding $\mathcal{C}$ of our Locally Decodable Code encodes a message $m = (m_1, \ldots, m_k)$ by

$$m \mapsto \mathcal{C}(\sum_{i=1}^{k} m_i b_i)\ .$$

In order to prove Theorem 3.2.1 we show that the following algorithm is a c-smooth decoder (see Definition 3.1.19).

> **Input:** An oracle access to $w \in \mathbb{F}^X$ and an index $i \in \{1, \ldots, k\}$. Let $D_i = D * h_i = \sum_{j=1}^{q} c_j \cdot g_j h_i$.

[2] Note that this is the only place where we use the irreducibility of (ρ, V). We discuss it later in Section 3.4.

1. Set $y = \mathcal{C}(\rho(D_i)b_i) \in \mathbb{F}^X$. Pick $r \in X$ at random from the support of y.
2. For $j = 1, \ldots, q$ query w at location: $(g_j h_i)^{-1} \cdot r \in X$.
3. Calculate $n_i = \sum_{j=1}^{q} c_j w[(g_j h_i)^{-1} \cdot r]$.
4. Return $m_i = y[r]^{-1} n_i$.

In order to show that this algorithm is a c-smooth decoder we need to show that:

- Completeness, i.e., if $w = \mathcal{C}(\sum m_i b_i)$ then the algorithm returns m_i on input i.
- Smoothness, i.e., each query is uniformly distributed over a domain of size $c|X|$.

Completeness: Recall that by definition of the permutational representation it holds that $\tau(g)w[r] = w[g^{-1}r]$. Thus n_i (line 3 of the decoding algorithm) is equal to

$$n_i = \sum_{j=1}^{q} c_j w[(g_j h_i)^{-1} \cdot r] = (\tau(D_i)w)[r] \; .$$

Let us substitute $w = \mathcal{C}(\sum_j m_j b_j)$ in this equation.

$$n_i = (\tau(D_i)w)[r] = (\tau(D_i)\mathcal{C}(\sum_{j=1}^{k} m_j b_j))[r] \stackrel{1}{=} \mathcal{C}\left(\rho(D_i)\sum_{j=1}^{k} m_j b_j\right)[r] \tag{3.2.1}$$

$$= \sum_{j=1}^{k} m_j \mathcal{C}(\rho(D_i)b_j)[r] \stackrel{2}{=} m_i \mathcal{C}(\rho(D_i)b_i)[r] \; .$$

Here Equality 1 holds since $\mathcal{C}$ is a homomorphism of the representations ρ and τ and Equality 2 follows from Lemma 3.2.2.

Thus from the definition of y it follows that $n_i = m_i y[r]$. Therefore, the algorithm returns a correct answer at line 4.

Smoothness: Note that if r is uniformly distributed over a domain of size $c|X|$, then so is $g_j h_i \cdot r$. Thus we need to prove that r is uniformly distributed over a domain of size $c|X|$, This is equivalent to say that the support of y is of size $c|X|$.

Since $\rho(D)$ is of rank one it holds that $\mathrm{Im}\,\mathcal{C} \cdot \rho(D)$ is one dimensional. Therefore, from Condition 2 if follows that for every non-zero vector in $\mathrm{Im}\,\mathcal{C} \cdot \rho(D)$ has support of size at least $c|X|$. Note that $y = \mathcal{C}(\rho(D * h_i)b_i) = \mathcal{C} \cdot \rho(D)(\rho(h_i)b_i)$. Thus $y \in \mathrm{Im}\,\mathcal{C} \cdot \rho(D)$ and from Lemma 3.2.2 it follows that $y \neq 0$.

□

3.2.1 Example: Two Query LDC from Representations of S_n

The goal of this subsection is to give a concrete example of irreducible representation which allows to construct two query LDC from Theorem 3.2.1. We want to mention that Hadamard Code can be captured by generalization of Theorem 3.2.1 see Section 3.4.2 for more details. The example given in this section has slightly worse parameters, but it is much simpler.

Let $\mathbb{F}$ be any algebraically closed field. The group S_n has a natural action on $[n]$. This action defines representation ρ on $\mathbb{F}^n$. This representation decomposes into a trivial representation which is spanned by vector of all ones and its complement which is the set of all vectors with sum zero. Let V be this representation, i.e., $V = \{v \in \mathbb{F}^n | \sum_{i=1}^n v[i] = 0\}$. One can show that this is indeed an irreducible representation. We denote it by ρ_1. Consider $f = id - (1,2) \in \mathbb{F}[S_n]$ then we claim that rank of $\rho_1(f)$ is one. Indeed let $\vec{x} = (x_1, x_2, \ldots, x_n) \in V$ then $\rho_1(f)(x_1, x_2, \ldots, x_n) = (x_1, x_2, \ldots, x_n) - (x_2, x_1, x_3, \ldots, x_n) = (x_1 - x_2, x_2 - x_1, 0, \ldots, 0) \in V$. Thus $\operatorname{Im}\rho_1(f)$ is $\lambda(1, -1, 0, 0, \ldots, 0)$. Therefore $\operatorname{Rank}(\rho_1(f)) = 1$. Theorem 1.2.1 gives us immediately a 2 query $[n-1, n!]$ LDC.

Now let us show that using different sets X on which group S_n acts we can achieve tradeoff rate/soundness. Now let X be the set of all subsets of $[n]$ of size k. Then there exist a natural action of S_n on X which gives us permutational representation $\mathbb{F}^X$ (we think of $\mathbb{F}^X$ as all functions from subsets of size k to $\mathbb{F}$). Let us define $\mathcal{C}(x_1, x_2, \ldots, x_n) = g$ where g is a function which takes subset of size k as input and outputs the sum of this subset, i.e., g defined by

$$g(S) = \sum_{j \in S}^{k} x_j.$$

We can see that the support of $\mathcal{C}(\rho_1(f)) = \mathcal{C}(1, -1, 0, \ldots, 0)$ is all subsets which contains exactly one of the elements: 1 or 2. The length of the code is $\binom{n}{k}$. The number of k-subsets that contain exactly one of the elements 1 or 2 is $2\binom{n-2}{k-1}$. Therefore, the relative support is $2\binom{n-2}{k-1}/\binom{n}{k} = \frac{2k(n-k)}{n(n-1)}$.

Using Theorem 3.2.1 we get two-query locally decodable codes $[n-1, \binom{n}{k}]$ with soundness $2\frac{k}{n}\frac{n-k}{n-1}$. This example shows that the parameters of the LDC depends not only on the representation it defines but also on the space in which we embed it into.

3.2.2 Embedding to the Regular Representation

Theorem 3.2.1 shows that in order to construct an LDC it is sufficient to do two things: First, construct an irreducible representation with a sparse rank one element. Second, embed it into a permutational representation such that the second condition of the theorem is satisfied. In this subsection we show that we can always embed any representation into the regular representation in a way that satisfies the second condition of the theorem.

Lemma 3.2.3. Let V be a vector space over a field $\mathbb{F}$. Then for every irreducible representation (ρ, V) and for every $v \in V,\ v \neq 0$ there exist a homomorphism $\mathcal{C} : V \to \mathbb{F}[G]$ of representations (ρ, V) and the regular representation in $\mathbb{F}[G]$ such that $\operatorname{supp}(\mathcal{C}(v)) \geq |G|(1 - \frac{1}{|\mathbb{F}|})$.

Proof. We view $\mathbb{F}[G]$ as a left representation of G. That is, $\tau(g)(f) = g * f$. For any $u \in V^*$ let us define a mapping $T_u : V \to \mathbb{F}[G]$ by:

$$T_u(x) = \sum_{g \in G} (\bar{\rho}(g)u(x))g. \tag{3.2.2}$$

We claim that T_u is an homomorphism from the representation (ρ, V) to regular representation

$\mathbb{F}[G]$. Indeed:

$$T_u(\rho(h)x) = \sum_{g\in G} \bar{\rho}(g)u(\rho(h)x)g = h * \sum_{g\in G} u(\rho(g^{-1}h)x)h^{-1}g.$$

Substituting $g = h^{-1}g$ we get

$$T_u(\rho(h)x) = h * \sum_{g\in G} u(\rho(g^{-1})x)g = \tau(h)T_u(x).$$

Now we want to show that for some $u \in V^*$ vector $T_u(v)$ has large support. Consider the set $U = \{T_u(v) : u \in V^*\}$. It is easy to see that it is a linear subspace of $\mathbb{F}[G]$ and that it have full support. From Lemma 3.1.5 it follows that there exists a vector with support at least $|G|(1 - \frac{1}{|\mathbb{F}|})$. Therefore, exist an u such that T_u is a G-homomorphism such that $T_u(v)$ has desired support. □

Note that when $\mathbb{F}$ is infinite field then we can get full support and all algebraically closed fields are infinite. As a corollary the last lemma we get Theorem 1.2.1.

Corollary 3.2.4 (Theorem 1.2.1)**.** Let V be a vector space over an algebraically closed field $\mathbb{F}$. Let G be a finite group and let (ρ, V) be an irreducible representation of G. Let $D \in \mathbb{F}[G]$ be an element of group algebra of sparsity q such that $\mathrm{Rank}(\rho(D)) = 1$. Then there exist locally $(q, \delta, q\delta)$ decodable code $\mathcal{C} : V \to \mathbb{F}^G$.

Assume that we have $(\rho, V), D \in \mathbb{F}[G]$ which satisfies the first condition of Theorem 3.2.1. Then from the corollary above it follows that we can embed (ρ, V) to the regular representation in a way that satisfies the second condition. A natural question to ask is can we embed it to a smaller permutational representation. The next lemma gives characterization of all such permutational representations.

Lemma 3.2.5. Let $(\rho, V), D \in \mathbb{F}[G]$ which satisfies the first condition of Theorem 3.2.1. Let $v \in \mathrm{Im}\,\rho(D)$ a non-zero vector and $H < G$ is any subgroup of G. Assume that exist $u \in V^*$ such that $\rho(h)u = u$ for every $h \in H$ and $|\{g \in G/H : u(\rho(g)v) \neq 0\}| \geq c|G/H|$ then there exist G-homomorphism $C : V \to \mathbb{F}^X$, where $X = G/H$, such that $\mathrm{supp}(C(v)) \geq c|X|$, i.e., it satisfies the second condition of Theorem 3.2.1.

Proof. Consider a subspace L_H of the regular representation $\mathbb{F}^G$ of functions constant on cosets of H, i.e., $L_H = \{f \in \mathbb{F}^G : \forall g \in G, \forall h \in H, f(gh) = f(g)\}$. It is easy to see that L_H is a sub-representation of the regular representation isomorphic to the permutational representation $\mathbb{F}^X$, where $X = G/H$. Let T_u be an embedding as in proof of Lemma 3.2.3 defined by Equation 3.2.2. Note that since $Hu = u$ for every $x \in V, h \in H_u$ it holds that $T_u(x)[g] = T_u(x)[gh]$, i.e., $T_u(x) \in L_H$. Therefore T_u is an embedding to the permutational representation $\mathbb{F}^X$, where $X = G/H_u$. From the definition of T_u it follows that the support of $T_u(v)$ is exactly $|\{g \in G/H_u : u(\rho(g)v) \neq 0\}|$. □

Remark 3.2.1. *It could be shown that any embedding satisfying second condition of Theorem 3.2.1 could be decribed by Lemma 3.2.5.*

3.2.3 Alphabet Reduction

In this section we show that we can transform codes over any algebraic extension of $\mathbb{F}_p$ to codes over $\mathbb{F}_p$. The reduction which we give here adapts the reduction from [12] to our settings.

Theorem 3.2.6. *Let $\mathbb{F}$ be a field of characteristic p and let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a $(q, \delta, \frac{q\delta}{c})$-LDC as in Theorem 3.2.1. Then there exist a code $\tilde{\mathcal{C}} : \mathbb{F}_p \to \mathbb{F}_p^{X\times[q]}$ which is $(q, \delta, \frac{p}{p-1}\frac{q\delta}{c})$-LDC.*

Proof. First let us rescale the basis so that we will have the same decoding vector for every message symbol. Let v be any vector in $\mathrm{Im}(\rho(D))$. Set $y = \mathcal{C}(v)$. Recall that in the proof of Theorem 3.2.1 we have showed that $\mathcal{C}(\rho(D_i)b_i) = \lambda_i y$. We can replace b_i with $\lambda_i^{-1}b_i$ so that $\mathcal{C}(\rho(D_i)b_i) = y$. It follows from the assumption that y has support $c|X|$. For a linear functional $\ell : \mathbb{F} \to \mathbb{F}_p$, we denote by $\ell(y)$ vector achieved by applying ℓ on each coordinate of y. From standard random argument there exists a linear functional ℓ such that support of $\ell(y)$ is at least $\frac{p-1}{p}c|X|$. Let us fix such an ℓ. Let $D = \sum_{i=1}^q c_i g_i \in \mathbb{F}[G]$ be a rank one element. Let us define $\tilde{\mathcal{C}}$ by $\tilde{\mathcal{C}}(m)[x, i] = \ell(c_i\mathcal{C}(m)[x])$. We need to show that this is an LDC. Let us describe the decoding algorithm:

Input: An oracle access to $w \in \mathbb{F}^X$ and bit index i.
Let $D_i = D*h_i = \sum_{j=1}^q c_j \cdot g_j h_i$ be where $h_i \in G$ is a group element as in Lemma 3.2.2.

1. Set $y = \mathcal{C}(\rho(D_i)b_i) \in \mathbb{F}^X$. Pick r at random from the support of $\ell(y)$.
2. For $j = 1, \ldots, q$ query w at location: $((g_j h_i)^{-1} \cdot r, j)$.
3. Calculate $\tilde{n}_i = \sum_{j=1}^q w[(g_j h_i)^{-1} \cdot r, j]$.
4. Return $m_i = \ell(y[r])^{-1}\tilde{n}_i$.

Now let us show that this decoding algorithm returns the correct answer when it receives an uncorrupted codeword. If $w = \tilde{\mathcal{C}}(m)$, then

$$\tilde{n}_i = \ell(\sum_{j=1}^{q} c_j \mathcal{C}((g_j h_i)^{-1} r)) = \ell(\tau(D_i)\mathcal{C}(m)[r]).$$

Recall that from Equation 3.2.1 it follows that $\tau(D_i)\mathcal{C}(m)[r] = m_i y[r]$. Thus we get that $\tilde{n}_i = m_i \ell(y[r])$. Thus, the decoding algorithm returns the correct answer on line 4 on an uncorrupted codeword.

Now let us prove that if $\tilde{\mathcal{C}}(m)$ is corrupted in at most δ coordinates, then the decoder reads a corrupted place with probability at most $\frac{p}{p-1}\frac{q\delta}{c}$. Let us call the coordinates of type (x, i) the i^{th} block. Let δ_i proportion of coordinates i^{th} block which are corrupted, and notice that $\sum \delta_i = q\delta$. Note that i^{th} query is distributed uniformly over $\frac{p-1}{p}c$ fraction of coordinates of the i^{th} block. Therefore, the probability that i^{th} coordinate is corrupted is $\frac{p}{p-1}\frac{\delta_i}{c}$. Thus by union bound we get that at least one of the coordinates is corrupted with probability at most $\sum \frac{p}{p-1}\frac{\delta_i}{c} = \frac{p}{p-1}\frac{q\delta}{c}$. □

We have the following corollary from this theorem and Theorem 1.2.1.

Corollary 3.2.7. Let $\mathbb{F}$ be a field of characteristic p. Let G be a finite group and let (ρ, V) be an irreducible representation of G and let $k = \dim V$. Let $D \in \mathbb{F}[G]$ be an element of

group algebra of sparsity q such that $\mathrm{Rank}(\rho(D)) = 1$. Then there exist $(q, \delta, \frac{p}{p-1}q\delta)$-LDC $\mathcal{C} : \mathbb{F}_p^k \to \mathbb{F}_p^{G\times[q]}$.

3.3 Matching Vector Codes and Abelian Invariant Codes

In the next two sections we show that there exists irreducible representations such that Theorem 3.2.1 gives codes matching the parameters of [12]. We show that the codes constructed in [12] could be interpreted as a construction of an irreducible representation.

In this section we show that if MV is an orbit of a group H then one can construct from such MV an irreducible representation with a sparse element in the group algebra of rank one. In the next section, we show that the variant of the Grolmusz's [23] construction described in [12],[3] is MV that is an orbit of the symmetric group.

Let A be an Abelian group. Recall that *the dual group* A^* is the set of all group homomorphisms $v : A \to \mathbb{Z}_m$, where m is the order of the group. In this section it will be more convenient for us to work with the following generalization of MV to any Abelian group:

Definition 3.3.1. *Let A be an Abelian group. Let m be the order of A. For any set $S \subset \mathbb{Z}_m - \{0\}$ the families $\mathcal{U} = \{u_i\}_{i=1}^k \subset A, \mathcal{V} = \{v_i\}_{i=1}^k \subset A^*$ are* S-Matching Vectors (MV) *if the following conditions hold:*

1. $v_j(u_i) \in S$ *for every* $i \neq j$.
2. $v_i(u_i) = 0$ *for every* $i \in [k]$.

Note that if $A = \mathbb{Z}_m^h$ using the isomorphism $\psi : A \to A^*$ given by $\psi(v)(x) = \langle v, x\rangle$ we get the standard definition of MV.

In this section we assume that the characteristics of $\mathbb{F}$ is co-prime to m and that there exists $\gamma \in \mathbb{F}^*$ an element of order m, i.e., $\gamma^m = 1$ and $\gamma^i \neq 1$ for $0 < i < m$. Then for any $v \in A^*$ we denote by γ^v the function from A to $\mathbb{F}$ defined by $\gamma^v(a) = \gamma^{v(a)}$. Let H be any *group that acts on the group* A. (Recall Definition 3.1.5 of an action of a group on a group.) In this case H also acts on A^*, where an action is given by the rule: $(h \cdot v)(x) = v(h^{-1} \cdot x)$. The group $G = A \rtimes H$ by Definition 3.1.6 acts on the set A. Let $(\tau, \mathbb{F}^A)$ be the corresponding permutational representation of G.

Definition 3.3.2. *A polynomial $p(x) \in \mathbb{F}[x]$ is S-decoding if $p(\gamma^s) = 0$ for all $s \in S$ and $p(1) = 1$.*

The goal of this section is to prove the following theorem:

Theorem 3.3.1. *Let $\mathcal{U}, \mathcal{V}$ be S-Matching Vectors such that $\mathcal{V}$ is an orbit of H. Let $p(x)$ be an S-decoding polynomial of sparsity q. Then there exists an irreducible representation (ρ, L), a permutational representation $(\tau, \mathbb{F}^A)$ of $G = A \rtimes H$ and $D = \sum_{i=1}^q c_i g_i$ which satisfy the conditions of Theorem 3.2.1 with $c = 1, \dim L = |\mathcal{V}|$.*

Proof. First we need to construct a representations (L, ρ). We do it in next two lemmas.

[3] Using the same ideas it is also possible to prove the statement for Grolmusz's construction.

Lemma 3.3.2. For any $\mathcal{V} \subset A^*$, the vector space $L \subset \mathbb{F}^A$ defined by

$$L = \mathrm{Span}\{\gamma^v : v \in \mathcal{V}\} \subset \mathbb{F}^A \tag{3.3.1}$$

is a sub-representation of the regular representation of group A of dimension $|\mathcal{V}|$.

Proof. First let us show that L is closed under action of A. For any $v \in A^*$ it holds that:

$$\gamma^v(a+b) = \gamma^{v(a+b)} = \gamma^{v(a)+v(b)} = \gamma^{v(a)}\gamma^{v(b)} = \gamma^v(a)\gamma^v(b).$$

Thus γ^v is a one-dimensional sub-representation of the regular representation of the group A. Therefore, L is a sub-representation. Note that for $v_1 \neq v_2$, the representations $\gamma^{v_1}, \gamma^{v_2}$ are non isomorphic one-dimensional sub-representations. Therefore, from Lemma 3.1.4 it follows that $\{\gamma^v\}_{v\in\mathcal{V}}$ are linearly independent vectors. Thus the dimension of L is $|\mathcal{V}|$. □

Lemma 3.3.3. For any $\mathcal{V} \subset A^*$ closed under action of H it holds that the vector space L defined by Equation 3.3.1 is a sub-representation of the permutational representation $(\tau, \mathbb{F}^A)$ of the group $G = A \rtimes H$.

Proof. Let $v \in A^*$ and consider the vector $\gamma^v \in \mathbb{F}^A$. Then for $h \in H$ it holds that

$$(\tau(h)\gamma^v)(x) = \gamma^{v(h^{-1}x)} = \gamma^{h\cdot v}(x)\ .$$

Thus if $\mathcal{V}$ is closed under the action of H then the vector space L is closed under the action of $\tau(h)$ for $h \in H$. Since L is a representation of A, it is also closed under the action of $\tau(a)$ for $a \in A$. Since H and A generate G, the space L is a sub-representation of $(\tau, \mathbb{F}^A)$. □

Let us denote this sub-representation by (ρ, L) and by $\mathcal{C} : L \to \mathbb{F}^A$ its embedding into $\mathbb{F}^A$. Note that $\dim L = |\mathcal{V}|$.

Lemma 3.3.4. If H acts transitively on $\mathcal{V}$ then the representation (ρ, L) of G is irreducible.

Proof. Assume that $\tilde{L} \subset L$ is a non-zero sub-representation of L. In order to prove that (ρ, L) is an irreducible representation we need to prove that $\tilde{L} = L$. Now let us look on L and $\tilde{L}$ as a representations of the group A. Then $L = \oplus_{v\in\mathcal{V}}\gamma^v$ is a direct sum of non-isomorphic irreducible representations. Therefore there exists some subset $\tilde{\mathcal{V}} \subset \mathcal{V}$ such that $\tilde{L} = \oplus_{v\in\tilde{\mathcal{V}}}\gamma^v$. In particular for some $v \in \mathcal{V}$ it holds that $\gamma^v \in \tilde{L}$. Since H acts transitively on $\mathcal{V}$ for every $v' \in \mathcal{V}$ there exists $h \in H$ such that $h \cdot v = v'$. Thus it holds that

$$\tau(h)\gamma^v = \gamma^{h\cdot v} = \gamma^{v'}\ .$$

Therefore, we proved that:

$$L = \mathrm{Span}\{\gamma^v : v \in \mathcal{V}\} \subset \tilde{L} \subset L.$$

Thus $\tilde{L} = L$. □

Let $p(x) = \sum_{i=1}^q c_i x^{t_i}$ be the given S-decoding polynomial. We define D as $D = \sum_{i=1}^q c_i(-t_i u_1)$, where we think of $-t_i u_1$ as an element of G. We claim that $\mathrm{Rank}\,\rho(D) = 1$.

Note that the set $\{\gamma^{v_i}\}_{i=1}^k$ forms a basis of L. Let us show that $\rho(D)\gamma^{v_i} = 0$ for $i \neq 1$ and $\rho(D)\gamma^{v_1} = \gamma^{v_1}$. Indeed:

$$\rho(D)\gamma^{v_i} = \sum c_i \rho(-t_i u_1)\gamma^{v_i} = \gamma^{v_i} \sum c_i \gamma^{v_i(t_i u_1)} = \gamma^{v_i} \sum c_i (\gamma^{v_i(u_1)})^{t_i} = \gamma^{v_i} p(\gamma^{v_i(u_1)}).$$

Since $v_i(u_1) \in S$ for all $i \neq 1$ it follows that $p(\gamma^{v_i(u_1)}) = p(\gamma^s)$ for some $s \in S$ and since p is an S-decoding polynomial it is equal to 0. Thus $\rho(D)\gamma^{v_i} = 0$ for $i \neq 1$ and for $i = 1$

Note that for natural embedding of $\mathcal{C} : L \to \mathbb{F}^A$ it holds that $\operatorname{Im}\mathcal{C}(\rho(D)) = \operatorname{Span}\{\gamma^{v_1}\}$ has full support. Therefore, smoothness constant c in Theorem 3.2.1 is 1. □

Remark 3.3.1. *The representation of (ρ, L) defined in the proof is: $\mathrm{Ind}_{A\rtimes F}^G \gamma^v$, where v is any element in $\mathcal{V}$ and $F = \{h \in H : h \cdot v = v\}$ be a subgroup of H.*

Note that in the proof of the above theorem we used only one element u_1 of $\mathcal{U}$. The following lemma shows that if $\mathcal{V}$ is an orbit of some group and "matching" one element then we can construct $\mathcal{U}$ to be orbit of the same group such that $\mathcal{U}, \mathcal{V}$ are Matching Vectors.

Lemma 3.3.5. Let $\mathcal{V} = \{h_i \cdot v\}_{i=1}^k \subset A^*$ be an orbit of H such that for some $u \in A$ it holds that $h_1 v(u) = 0$ and $h_i v(u) \in S$ for $i \neq 1$. Then the family $\mathcal{U} = \{h_i u\}_{i=1}^k, \mathcal{V} = \{h_i \cdot v\}_{i=1}^k$ is a family of S-Matching Vectors.

Proof. First note that $h_i v(h_i u) = h_i^{-1} h_i v(u) = v(u) = 0$. Next for $i \neq j$ it holds that $h_i v(h_j u) = h_j^{-1} h_i v(u)$. Since $\mathcal{V}$ is an orbit there exist k such that $h_j^{-1} h_i v = h_k v$, $k \neq 1$ since $i \neq j$. Therefore $h_i v(h_j u) = h_k(u) \in S$. □

3.4 Is Irreducibility Essential?

Representation theory when characteristic of the field divides the size of the group called modular representation theory. Modular representation theory is very different from non-modular case. In this section we ask the question does irreducibility in Theorem 3.2.1 essential. We show that in non-modular case the answer is Yes[4]. We show that in modular case we can construct reducible representation which will lead to LDC. Thus we can see Theorem 3.4.1 as a generalization of the Theorem 3.2.1 to modular representation theory.

It may happen that some representation is irreducible over field $\mathbb{F}$, but reducible over algebraic closure of $\mathbb{F}$. Representations which are irreducible over algebraic closure of $\mathbb{F}$ called completely irreducible. Although for the proof of the Theorem 3.2.1 we do not need complete irreducibility we show that in order to have rank one element complete irreducibility is essential for non-modular representations.

In the proof of Theorem 3.2.1, the only reason why we need the fact that (ρ, V) is irreducible is to show that the orbit of u spans all the dual space. Therefore, we can make the following generalization of Theorem 3.2.1:

Theorem 3.4.1. *Let G be a finite group. Let (ρ, V) be any representation of G, $(\tau, \mathbb{F}^X)$ be a permutational representation of G. Let $\mathcal{C} : V \to \mathbb{F}^X$ be a G-embedding. Assume that the following conditions hold:*

[4] In fact we show that the representation should be indecomposable. In non-modular case all indecomposable representations are irreducible.

1. (a) *There exists a q-sparse element $D \in \mathbb{F}[G]$, $D = \sum_{i=1}^{q} c_i g_i$ such that* $\mathrm{Rank}(\rho(D)) = 1$.

   (b) *Let $u \in V^*$ be a non-zero linear functional such that* $\mathrm{Ker}\, u = \mathrm{Ker}\, \rho(D)$. *Then the set $\{\bar{\rho}(g)u | g \in G\}$ spans V^*.*

2. $\mathrm{Im}(\mathcal{C} \circ \rho(D))$ *has a support $c|X|$.*

Then there exists a basis $b_1, \ldots, b_k$ for V such that $\mathcal{C}(\sum(m_i b_i))$ is $(q, \delta, \frac{q\delta}{c})$-LDC.

From Lemma 3.1.3 it follows that irreducibility of the representation (ρ, V) implies Condition 1b of this theorem. Here we show that if characteristics of the field $\mathbb{F}$ is does not divides $|G|$ then the converse is also true, i.e., if u spans dual space V^* then (ρ, V) is irreducible.

3.4.1 Yes!

Theorem 3.4.2. *Let V be a vector space over an algebraically closed field $\mathbb{F}$ of characteristic which does not divides $|G|$. Let ρ be a representation of group G in the vector space V. Let $f \in \mathbb{F}[G]$ such that* $\mathrm{Rank}\, \rho(f) = 1$. *Let $u \in V^*$ such that* $\mathrm{Ker}\, u = \mathrm{Ker}\, \rho(f)$. *If $V^* = \mathrm{Span}\{\bar{\rho}(g)u | g \in G\}$, then V is an irreducible representation.*

Proof. Let us assume by contradiction that V is reducible. Then from Theorem 3.1.2 it follows that $V = V_1 \oplus V_2$. This mean that in basis of V_1 and V_2 for every $g \in G$ the matrix $\rho(g)$ is of form

$$\rho(g) = \begin{pmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{pmatrix},$$

where ρ_1, ρ_2 restrictions of ρ to V_1, V_2. Therefore $\rho(f) = \sum a_i \rho(g_i)$ is of form

$$\rho(f) = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

But this matrix may be of rank one only if A or B is zero. Let us assume w.l.g. that B is zero. But then $V_2 \subset \mathrm{Ker}\, \rho(f)$. Therefore it holds that $u(V_2) = 0$. Since V_2 is invariant space, it also holds that $\bar{\rho}(g)u(V_2) = 0$. Since $\bar{\rho}(g)u$ span V^*, it must be that $V_2 = 0$. □

If V is a vector space over some field $\mathbb{F}$, then $\mathrm{Rank}\, \rho(f) = 1$ also over algebraic closure of $\mathbb{F}$ and thus (ρ, V) should be irreducible not just over $\mathbb{F}$, but also over the algebraic closure of $\mathbb{F}$.

3.4.2 No!

Let us now give an example of reducible representations (ρ, V) when vector u spans all the dual space. Of course in this example characteristics of $\mathbb{F}$ divides $|G|$. This example is a well known Reed-Muller Code. Let $\mathbb{F}$ be a field of characteristic $p \geq d$. Let us consider the group G of affine transformations over $\mathbb{F}_p^k$, i.e., $G = \{A\vec{x} + b : A \in \mathrm{GL}(p, k), b \in \mathbb{F}_p^k\}$. Let us set $X = \mathbb{F}_p^k$ and $(\tau, \mathbb{F}^X)$ be corresponding permutational representation of G. Let $\mathrm{RM}(d, k) \subset \mathbb{F}^X$ be a vector space of polynomials of total degree at most d with coefficients in $\mathbb{F}$. It is easy to verify that $\mathrm{RM}(d, k)$ is invariant under permutations of G. Thus $\mathrm{RM}(d, k)$ is a sub-representation

of $\mathbb{F}^X$. Let us denote it by $(\rho, \mathrm{RM}(d,k))$. Let $\mathrm{const} \subset \mathrm{RM}(d,k)$ be a subspace of constant functions. Note that const is a sub-representation of V. Thus $\mathrm{RM}(d,k)$ is reducible. Let us pick $\lambda \neq 1 \in \mathbb{F}_p$ be a generator of the $\mathbb{F}_p^*$. Let $m_\lambda \in G$ be a permutation $\vec{x} \mapsto \lambda\vec{x}$.

Lemma 3.4.3. There exists $c_0, c_2, \ldots, c_d$ such that the following holds: Let $D = \sum c_i m_\lambda^i \in \mathbb{F}[G]$ then the mapping $\rho(D)$ is of rank one and given a polynomial $p \in \mathbb{F}^X$ the mapping $\rho(D)$ returns a constant function $p(\vec{0})$.

Proof. Let us consider how $\rho(m_\lambda)$ acts on p. Let $p = \sum_{j=0}^{d} p_j$, where p_j is a homogeneous part of p of degree j. Then it holds that

$$\rho(m_\lambda)p(x) = p(\lambda^{-1}x) = \sum_{j=0}^{d} \lambda^{-j} p_j(x) \; .$$

In the same way it for every i it also holds that

$$\rho(m_\lambda^i)p(x) = p(\lambda^{-i}x) = \sum_{j=0}^{d} \lambda^{-ij} p_j(x) \; . \tag{3.4.1}$$

Let $V[i,j] = \lambda^{-ij}$ be a Vandermonde matrix. For vector $\vec{c} = (c_0, c_2, \ldots, c_d)$ let $a = (a_0, \ldots, a_d) = V \cdot \vec{c}$. Then from Equation 3.4.1 it follows that:

$$\rho(\sum_{i=0}^{d} c_i m_\lambda^i)p = \sum_{i=0}^{d} a_i p_i \; . \tag{3.4.2}$$

Note that V is invertible matrix. Thus we can choose $\vec{c}$ such that $V \cdot \vec{c} = (1, 0, \ldots, 0)$. Substituting this $\vec{c}$ in Equation 3.4.2 we get:

$$\rho(\sum_{i=0}^{d} c_i m_\lambda^i) = p_0 \; .$$

But p_0 is a constant term of p which exactly equal to $p(\vec{0})$. □

Now consider a linear functional $u : \mathrm{RM}(d,k) \to \mathbb{F}$ given by $u(p) = p(\vec{0})$. Then definitely it holds that $\mathrm{Ker}\, u = \mathrm{Ker}\, \rho(D)$.

Lemma 3.4.4. Then the set $\{\bar{\rho}(g)u | g \in G\}$ spans the dual space of $\mathrm{RM}(d,k)$.

Proof. Note that linear functionals $u_1, u_2, \ldots, u_k$ span the dual space iff $\cap_{i=1}^{k} \mathrm{Ker}\, u_i = 0$. For $b \in \mathbb{F}_p^k$ let $g_b \in G$ be a permutation $x \mapsto x + b$. Let us show that:

$$\cap_{b \in \mathbb{F}_p^k} \mathrm{Ker}\, g_b u = 0 \; .$$

Indeed $g_b u(p) = p(b)$. Thus if $p \in \cap_{b \in \mathbb{F}_p^k} \mathrm{Ker}\, g_b u$ then $p(b) = 0$ for every $b \in \mathbb{F}_p^k$. Thus it must be that $p = 0$. □

3.5 G-Invariant Codes and Representations of G

In this section we show tight connections between linear G-invariant codes and the representations of the group G. We show that there exists a one-to-one correspondence between sub-representations of the permutational representations and G-invariant codes. Furthermore we can define a representation in the message space so that the code becomes a G-homomorphism.

Let us first define G-invariant codes:

Definition 3.5.1. *Let G be a group acting on a set $X = \{x_i\}_{i=1}^n$. Let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a code. We say $\mathcal{C}$ is G-invariant iff for every $c = (c_{x_1}, c_{x_2}, \ldots, c_{x_n}) \in \mathrm{Im}(\mathcal{C})$, and for every $g \in G$ it holds that*

$$g \cdot c = (c_{g^{-1}\cdot x_1}, c_{g^{-1}\cdot x_2}, \ldots, c_{g^{-1}\cdot x_n}) \in \mathrm{Im}(\mathcal{C}) \; .$$

The action of G on X defines a permutational representation $(\tau, \mathbb{F}^X)$ of G (see Equation 3.1.2). We claim that there is a one to one correspondence between linear G-invariant codes and sub-representations of $(\tau, \mathbb{F}^X)$.

Lemma 3.5.1. Let G be a group that acts on the set X. Let $(\tau, \mathbb{F}^X)$ be the permutational representation defined by this action. Let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a linear code. Then $\mathcal{C}$ is G-invariant if and only if $\mathrm{Im}\,\mathcal{C}$ is a sub-representation of $(\tau, \mathbb{F}^X)$.

Proof. The proof almost follows from the definition. Let $c \in \mathrm{Im}\,\mathcal{C}$, where $c = (c_{x_1}, c_{x_2}, \ldots, c_{x_n})$ and consider c as a function from X to $\mathbb{F}$. Then $(g \cdot c)(x) = c(g^{-1}x)$ and by the definition of τ we have that $(\tau(g)c)(x) = c(g^{-1}x)$. Thus the code $\mathcal{C}$ is G-invariant if and only if for every $c \in \mathrm{Im}(\mathcal{C})$ and for every $g \in G$ it holds that $\tau(g)c \in \mathcal{C}$ i.e., if and only if $\mathrm{Im}\,\mathcal{C}$ is a sub-representation of $(\tau, \mathbb{F}^X)$. □

As a corollary we get that G-homomorphisms into permutational representations are G-invariant codes.

Corollary 3.5.2. Let G be a group acting on X. Let $(\tau, \mathbb{F}^X)$ be the permutational representation defined by this action. Let $(\rho, \mathbb{F}^k)$ be any representation of G. Let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a homomorphism of the representations $(\rho, \mathbb{F}^k)$ and $(\tau, \mathbb{F}^X)$ then $\mathcal{C}$ is a G-invariant code.

Proof. This follows from Lemma 3.5.1 and the fact that image of a G-homomorphism is a sub-representation. □

Let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a linear one-to-one G-invariant code. We already know that $\mathrm{Im}(\mathcal{C})$ is a sub-representation of $(\tau, \mathbb{F}^X)$. Let us show that we can define a representation $(\rho, \mathbb{F}^k)$ such that $\mathcal{C}$ is a G-homomorphism.

Theorem 3.5.3. *Let G be a group acting on X. Let $(\tau, \mathbb{F}^X)$ be the permutational representation defined by this action. Let $\mathcal{C} : \mathbb{F}^k \to \mathbb{F}^X$ be a linear one-to-one G-invariant code. Define a representation ρ of G in $\mathbb{F}^k$ by:*

$$\rho(g)(v) = \mathcal{C}^{-1}(\tau(g)\mathcal{C}(v)) \; . \tag{3.5.1}$$

Then $\mathcal{C}$ is an embedding of the representations $(\rho, \mathbb{F}^k)$ in $(\tau, \mathbb{F}^X)$.

Proof. First we need to proof that $\rho(g)$ is well defined: Since $\mathcal{C}$ is one-to-one $\mathcal{C}^{-1}$ is defined on $\operatorname{Im}\mathcal{C}$. Since $\mathcal{C}$ is closed under G it holds that $\tau(g)\mathcal{C}(v) \in \mathcal{C}$ therefore $\mathcal{C}^{-1}$ is defined on $(\tau(g)\mathcal{C}(v))$.

Now let us show that $\mathcal{C}$ is a G homomorphism:

$$\mathcal{C}(\rho(g)v) = \mathcal{C}(\mathcal{C}^{-1}(\tau(g)\mathcal{C}(v))) = \tau(g)\mathcal{C}(v) \ .$$

□

Chapter 4

Amplifying the Error-Tolerance of Locally Decodable Codes

4.1 Definitions

The *agreement* between strings x and y is the fraction of coordinates i in which $x_i = y_i$. The agreement between x and y is denoted by $\mathrm{Ag}(x, y)$.

Definition 4.1.1. *A probabilistic oracle machine* M^w locally outputs a string s with confidence $1 - \epsilon$, *if*

$$\forall i \ \Pr[M^w(i) = s_i] \geq 1 - \epsilon,$$

where the probability is over the randomness of M.

Definition 4.1.2. *A deterministic oracle machine* M^w locally ϵ-approximates a string s , *if*

$$\Pr_i[M^w(i) = s_i] \geq 1 - \epsilon,$$

where the probability is over a uniformly chosen i.

Note that if M^w locally outputs a string s with confidence $1 - \epsilon$ then there is a way to fix its randomness such that it will locally ϵ-approximates a string s. Essentially, M^w locally approximates a string s if it outputs a string that is close to s.

Definition 4.1.3 (Local unique decoding)**.** *A code* $\mathcal{C} : \Sigma^n \mapsto \Sigma^{\bar{n}}$ *is* (q, ϵ, δ) locally decodable *if there exists a probabilistic oracle machine* M^w *(the decoding algorithm) with oracle access to a received codeword* w *such that:*

1. *For every message* $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Sigma^n$ *and for every* $w \in \Sigma^{\bar{n}}$ *such that* $\mathrm{Ag}(\mathcal{C}(\lambda), w) \geq 1 - \delta$ *it holds that* M^w *locally outputs* λ *with confidence* $1 - \epsilon$.

2. $M^w(i)$ *makes at most* q *queries to* w *for all* $i \in [n]$.

It is possible to consider a more relaxed notion of local decoding, where the machine M^w is not required to successfully decode every i. Instead, it is required to succeed on average over i:

Definition 4.1.4 (Approximate local unique decoding)**.** *A code $\mathcal{C}$ over a field Σ, $\mathcal{C} : \Sigma^n \mapsto \Sigma^{\bar{n}}$ is* (q, ϵ, δ) approximately locally decodable *if there exists a deterministic oracle machine M^w (the decoding algorithm) with oracle access to a received codeword w such that:*

1. *For every message $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Sigma^n$ and for every $w \in \Sigma^{\bar{n}}$ such that $\mathrm{Ag}(\mathcal{C}(\lambda), w) \geq 1 - \delta$, it holds that M^w locally ϵ-approximates λ.*
2. *$M^w(i)$ makes at most q queries to w for all $i \in [n]$.*

Although the definitions of locally decodable codes and approximately locally decodable codes are similar, it appears that it is much harder to construct locally decodable codes than approximately locally decodable codes. While there exist constant-query approximately locally decodable codes of polynomial length, no such locally decodable codes are known. Approximately locally decodable are interesting when $\epsilon < \delta$, since the identity code is a $(1, \epsilon, \epsilon)$ approximately locally decodable code.

A code $\mathcal{C}$ is list-decodable if for every word, there are a few codewords near it. Let $\mathcal{C}(y_1), \mathcal{C}(y_2), \ldots, \mathcal{C}(y_L)$ be the list of codewords near a word w. Roughly speaking, a code $\mathcal{C}$ is locally list-decodable if there exists a machine M, that given i, j and an oracle access to the received word w, outputs the jth symbol of y_i. The locality property requires that the machine M makes a few queries to w. Formally:

Definition 4.1.5 (Local list-decoding)**.** *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ be a code. A set of probabilistic oracle machines $M_1, \ldots, M_L$ with oracle queries to w,* (α, L, q, ϵ) locally list-decodes *$\mathcal{C}$ at the word $w \in \Sigma^{\bar{n}}$, if,*

- *Every oracle machine M_j makes at most q queries to the input word w.*
- *For every codeword $c \in \mathcal{C}$ with $\mathrm{Ag}(c, w) \geq \alpha$, there exists some $k \in [L]$, such that M_k^w locally outputs c with confidence $1 - \epsilon$.*

We can also define *approximate* local list-decoding by relaxing the requirement that M_k^w successfully decodes c on every i. Instead, we require successful decoding on average over i.

Definition 4.1.6 (Approximate local list-decoding)**.** *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ be a code. A set of deterministic oracle machines $M_1, \ldots, M_L$ with oracle queries to w,* (α, L, q, ϵ) approximately locally list-decodes *$\mathcal{C}$ at the word $w \in \Sigma^{\bar{n}}$, if,*

- *Every oracle machine M_j makes at most q queries to the input word w.*
- *For every codeword $c \in \mathcal{C}$ with $\mathrm{Ag}(c, w) \geq \alpha$, there exists some $k \in [L]$, such that M_k^w ϵ-approximates c.*

Definition 4.1.7 ((Approximately) Locally list-decodable codes with deterministic reconstruction)**.** *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ be (α, L) list-decodable. A deterministic algorithm A (α, L, q, ϵ)* (approximately) locally list-decodes *$\mathcal{C}$, if on input n, A outputs oracle machines $M_1, \ldots, M_L$ which (α, L, q, ϵ) (approximately) locally list-decode $\mathcal{C}$ at* every *word $w \in \Sigma^{\bar{n}}$.*

The code $\mathcal{C}$ is (α, L) list-decodable and therefore every $w \in \Sigma^{\bar{n}}$ has at most L codewords $c_1, \ldots, c_L$ that are α-close to it. Each such codeword $c_i = \mathcal{C}(\lambda^i)$ is represented by a probabilistic machine M_i such that:

- If the code is locally list-decodable then $\forall j\ M_i(j) = \lambda^i_j$ with probability at least $1-\epsilon$.
- If the code is approximately locally list-decodable then $M_i(j) = \lambda^i_j$ for at least a $1-\epsilon$ fraction of the indices j.

The algorithm A outputs L machines that are good for every $w \in \Sigma^{\bar{n}}$. One way to think about it is that $i \in [L]$ is an advice that specifies which of the L solutions corresponds to the codeword we are interested in.

Definition 4.1.8 (Locally list-decodable codes with probabilistic reconstruction)**.** *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ be (α, L) list-decodable. A probabilistic algorithm A (α, L, q, ϵ)* locally list-decodes *$\mathcal{C}$, if on input n, A outputs probabilistic oracle machines $M_1, \ldots, M_L$ such that for every word $w \in \Sigma^{\bar{n}}$, with probability $2/3$ over the random coins of A, the machines $M_1, \ldots, M_L$ locally list-decode $\mathcal{C}$ at w, i.e.,*

$$\forall w \in F^{\bar{n}} \Pr_A \left[\forall \lambda \left(\mathrm{Ag}(\mathcal{C}(\lambda), w) \geq \alpha \ \Rightarrow\ \exists i\, \forall j \Pr[M_i(j) = \lambda_j] \geq 1-\epsilon\right)\right] \geq 2/3.$$

Definition 4.1.9 (Approximately locally list-decodable codes with probabilistic reconstruction)**.** *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$ be (α, L) list-decodable. A probabilistic algorithm A (α, L, q, ϵ)* approximately locally list-decodes *$\mathcal{C}$, if on input n, A outputs deterministic oracle machines $M_1, \ldots, M_L$ such that for every word $w \in \Sigma^{\bar{n}}$, with probability $2/3$ over the random coins of A, the machines $M_1, \ldots, M_L$ approximately locally list-decode $\mathcal{C}$ at w, i.e.,*

$$\forall w \in F^{\bar{n}} \Pr_A \left[\forall \lambda \left(\mathrm{Ag}(\mathcal{C}(\lambda), w) \geq \alpha \ \Rightarrow\ \exists i \Pr_j[M_i(j) = \lambda_j] \geq 1-\epsilon\right)\right] \geq 2/3.$$

The best approximately list-decodable codes currently known (to the best of our knowledge) are due to Impagliazzo et al. [26]. In this thesis we focus on binary codes, although by using the non-binary codes of [26] one can also get non-binary list-decodable codes.

Theorem 4.1.1 ([26][1])**.** *For every $\alpha, \epsilon > 0$ there exists a number $f(\alpha, \epsilon)$ such that there exists a code $\mathcal{C}_{\mathrm{App}} : \{0,1\}^n \mapsto \{0,1\}^{f(\alpha,\epsilon)n^5}$ which is $(1/2+\alpha, O(\frac{1}{\alpha^2}), O(\frac{\log(1/\epsilon)}{\alpha^3}), \epsilon)$ approximately locally list-decodable.*

4.2 Composition Theorem

Our main observation in this thesis is that if a code $\mathcal{C}_{\mathrm{LDC}}$ is locally decodable and a code $\mathcal{C}_{\mathrm{App}}$ is approximately locally decodable then by composing these two codes we get a code which is locally decodable, and can tolerate a higher error-rate.

Theorem 4.2.1. *Let $\mathcal{C}_{\mathrm{LDC}} : \Sigma_1^n \mapsto \Sigma_2^{N'}$ be (q, ϵ, δ) locally decodable code and let $\mathcal{C}_{\mathrm{App}} : \Sigma_2^{N'} \mapsto \Sigma_3^N$ be an (q', δ, δ') approximately locally decodable code. Then the code $\mathcal{C} = \mathcal{C}_{\mathrm{App}} \circ \mathcal{C}_{\mathrm{LDC}} : \Sigma_1^n \mapsto \Sigma_3^N$ defined by $\mathcal{C}(\lambda) = \mathcal{C}_{\mathrm{App}}(\mathcal{C}_{\mathrm{LDC}}(\lambda))$ is $(q \cdot q', \epsilon, \delta')$ locally decodable.*

[1] The code we use is not explicit in [26], but it can be deduced from Section 5 in that paper. In Section 5 it is shown that a longer code (the direct-product code, concatenated with Hadamard) is approximately locally list-decodable. However, the same proof carries over when using the derandomized direct-product code (concatenated with Hadamard). The parameter d (of [26]) is set to 5 (this is affects the exponent in the codeword length). The number of queries is $O(\frac{\log(1/\epsilon)}{\alpha^3})$ since we need to run the Goldreich-Levin algorithm $O(\frac{\log 1/\epsilon}{\alpha})$ times, and each run requires $1/\alpha^2$ queries.

Thus, if we have a locally decodable code which can tolerate a small fraction of errors, the above theorem allows us to amplify the error-rate by using an approximately locally decodable code. We have similar theorem for the list-decoding regime:

Theorem 4.2.2. *Let $\mathcal{C}_{\mathrm{LDC}} : \Sigma_1^n \mapsto \Sigma_2^{N'}$ be (q, ϵ, δ) locally decodable code and let $\mathcal{C}_{\mathrm{App}} : \Sigma_2^{N'} \mapsto \Sigma_3^N$ be an (α, L, q', δ) approximately locally list-decodable code. Then the code $\mathcal{C} = \mathcal{C}_{\mathrm{App}} \circ \mathcal{C}_{\mathrm{LDC}} : \Sigma_1^n \mapsto \Sigma_3^N$ defined by $\mathcal{C}(\lambda) = \mathcal{C}_{\mathrm{App}}(\mathcal{C}_{\mathrm{LDC}}(\lambda))$ is $(\alpha, q \cdot q', L, \epsilon)$ locally list-decodable.*

Proof. Let A denote the reconstruction algorithm for the code $\mathcal{C}_{\mathrm{App}}$ and let $D^w : [n] \mapsto \Sigma_1$ denote the unique decoding algorithm for the code $\mathcal{C}_{\mathrm{LDC}}$. The reconstruction algorithm for the code $\mathcal{C}$ works as follows: it first applies the algorithm A to obtain a list of machines $M_1, \ldots, M_L$. For each machine M_j, it outputs the machine Z_j defined by $Z_j^w(i) = D^{M_j^w}(i)$.

The bounds on the number of queries and the list size are immediate. Fix a word $w \in \Sigma_3^N$. The inner reconstruction algorithm A fails with probability at most $\frac{1}{3}$. When it does not fail, we will show that for every codeword with at least α agreement with w, its message is outputted with confidence $1 - \epsilon$ by one of the output machines. Suppose that the agreement between $\mathcal{C}_{\mathrm{App}}(\mathcal{C}_{\mathrm{LDC}}(\lambda))$ and w is at least α. Denote $\zeta = \mathcal{C}_{\mathrm{LDC}}(\lambda)$. Since A did not fail, one of the machines M_j^w δ-approximates ζ. Thus, $Z_j^w = D^{M_j^w}$ locally outputs λ with confidence $1 - \epsilon$. □

The above theorem give locally list-decodable codes which improve upon previously known constructions. Since we wish to get locally list-decodable codes with a constant query complexity, we need to use a locally decodable code with a constant query complexity. The best such codes currently known are due to [35]:

Theorem 4.2.3 ([35])**.** *For every $r \geq 2$ there exists a code*

$$\mathcal{C}_{\mathrm{LDC}} : \{0,1\}^n \mapsto \{0,1\}^{\exp(\exp(O(\sqrt[r]{\log n (\log\log n)^{r-1}})))}$$

which is $\left(\left(\frac{3}{4}\right)^{\min\{51, \lfloor r/2 \rfloor\}} 2^r, \gamma, 2 \cdot \left(\frac{3}{4}\right)^{\min\{51, \lfloor r/2 \rfloor\}} 2^r \cdot \gamma\right)$ locally decodable, for every $\gamma > 0$.

Let $\mathcal{C}_{\mathrm{LDC}}$ and $\mathcal{C}_{\mathrm{App}}$ be the codes from Theorem 4.2.3 and Theorem 4.1.1, respectively. Applying Theorem 4.2.2 on these codes gives:

Corollary 4.2.4 (Theorem 1.3.1 restated)**.** For every $r \geq 2$ and every $\alpha, \epsilon > 0$ there exists a code

$$\mathcal{C} : \{0,1\}^n \mapsto \{0,1\}^{f(\alpha, \frac{\epsilon}{2 \cdot 3^{r/2}}) \cdot \exp(\exp(O(\sqrt[r]{\log n (\log\log n)^{r-1}})))}$$

which is $(1/2 + \alpha, O(\frac{1}{\alpha^2}), O\left(\frac{r + \log(1/\epsilon)}{\alpha^3} \cdot 2^r\right), \epsilon)$ locally list-decodable, where f is the constant from Theorem 4.1.1.

Chapter 5

Open Problems

In this chapter we would like to summarize the most interesting open problems related to locally decodable codes.

5.1 Locally Decodable Codes

The first most obvious open problem is closing the gap between lower and upper bounds on the length of LDCs. For example, for 3-query LDCs the best known lower bound is $\Omega(k^2)$, while the best upper bound is only $\exp(\exp(\sqrt{\log k \log\log k}))$. In this thesis we have presented two approaches to improve upper bounds, namely, through matching vectors and through representation theory.

Matching Vectors A question of improving the parameters of matching vectors is a long standing problem. Recently it was shown in [9] that for a constant modulo one can not achieve polynomial rate matching vector codes[1]. We would like to mention that improving constructions of matching vectors will have consequences much beyond LDCs. For example, it may lead to the explicit Ramsey graphs.

Representaion Theory Constructing LDCs from the representation theory is a relatively new approach and thus we believe that it is more promising. We are not aware of any limitations on this model beyond the standard lower bounds for LDCs. It would be interesting if one could show any lower bounds on this model.

Locally Decodable Codes over $\mathbb{C}$ This question is not a main stream question, but we would like to mention it since we believe that the answer to this question will lead to a new insight in LDCs. In the Chapter 2 we have shown a generic way to construct sub-exponential 4-query LDCs which work over any field. Next, in order to reduce the number of queries to 3, we gave an example of S-decoding polynomial with 3 monomials. However this works only over field of characteristic 2. This can probably be extended to any finite characteristic, but it seems that it is impossible to construct such a S-decoding polynomial over a field with characteristic zero. Thus the question if one can construct 3 query linear LDCs over complex numbers of

[1] Assuming Polynomial Friemann Rusha conjecture

sub-exponential length is open. It is not obvious that such codes exist and there are some speculative reasons why such codes may not exist.

5.2 Self Correctable Codes

Self correctable codes are codes where instead of each message symbol, each codeword symbol could be corrected by reading a constant number of symbols. Note that self correctable codes must be LDCs, but not vise versa. For example, an Hadamard code is self correctable. Almost the only example of self correctable codes is the family of Reed-Muller codes. The question of constructing new families of self correctable codes is very interesting. We do not know yet if sub-exponential self correctable codes exist.

Proving lower bounds for self correctable codes seems to be easier than proving these bounds for LDCs. However we do not know better lower bounds for self correctable codes except when the number of queries is 2. For example, it was shown in [3] the impossibility of 2-query self correctable codes over $\mathbb{C}$. It was also shown in [8] that in case of 2-queries self correctable codes are longer than LDCs. Proving better lower bounds for self-correctable codes for 3 and more queries seems like a very challenging task.

Bibliography

[1] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. 1.3, 1.3

[2] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31. ACM, 1991. (document)

[3] Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications toc ombinatorial geometry and locally correctable codes. In *STOC*, pages 519–528, 2011. 5.2

[4] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. Comput. Syst. Sci.*, 71(2):213–247, 2005. (document)

[5] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the o(n1/(2k-1)) barrier for information-theoretic private information retrieval. In *FOCS*, pages 261–270, 2002. (document)

[6] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. Local list-decoding with a constant number of queries. In *FOCS*, 2010. 1.3, 1.3, 1.4

[7] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. A note on amplifying the error-tolerance of locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:134, 2010. (document), 1.4

[8] Arnab Bhattacharyya, Zeev Dvir, Amir Shpilka, and Shubhangi Saraf. Tight lower bounds for 2-query lccs over finite fields. In *FOCS*, pages 638–647, 2011. 5.2

[9] Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New lower bounds for matching vector codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:34, 2012. 1.1, 5.1

[10] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. Technical Report TR10-012, Electronic Colloquium on Computational Complexity (ECCC), 2010. 1.3

[11] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM J. Comput.*, 40(4):1154–1178, 2011. 1.1

[12] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *STOC*, pages 39–44, 2009. (document), 1.1, 1.1.1, 1.3, 1.3, 1.3, 3.2.3, 3.3

[13] Klim Efremenko. From irreducible representations to locally decodable codes, 2011. Unpublished manuscript. (document)

[14] Peter. Elias. List decoding for noisy channels. Technical report, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957. 1.3

[15] Anna Gal and Andrew Mills. Three query locally decodable codes with higher correctness require exponential length. 2009. 1.3

[16] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004. (document), 1, 1

[17] Oded Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005. (document), 1

[18] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *IEEE Conference on Computational Complexity*, pages 175–183, 2002. 1

[19] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006. (document)

[20] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989. 1.3, 1.3

[21] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM J. Discrete Math.*, 13(4):535–570, 2000. 1.3, 1.3

[22] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. List-decoding reed-muller codes over small fields. In *STOC*, pages 265–274, 2008. 1.3, 1.3

[23] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20(1):71–86, 2000. (document), 1.1, 1.1, 1.2, 2, 2.2.3, 2.2.1, 3.3

[24] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Approximately list-decoding direct product codes and uniform hardness amplification. In *FOCS*, pages 187–196, 2006. 1.3

[25] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. In *STOC*, pages 579–588, 2008. 1.3

[26] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM J. Comput.*, 39(4):1637–1665, 2010. 1.3, 3, 4.1, 4.1.1, 1

[27] Russell Impagliazzo and Avi Wigderson. = *BPP* if requires exponential circuits: Derandomizing the xor lemma. In *STOC*, pages 220–229, 1997. 1.3

[28] Toshiya Itoh and Yasuhiro Suzuki. New constructions for query-efficient locally decodable codes of subexponential length. *CoRR*, abs/0810.4576, 2008. 1.1, 2.2.3

[29] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000. (document), 1, 1

[30] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *FOCS*, pages 590–600, 2007. 1.3

[31] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *STOC*, pages 106–115. ACM, 2003. 1

[32] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004. (document)

[33] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of sparse random linear codes from high-error. Technical Report 115, Electronic Colloquium on Computational Complexity (ECCC), 2009. 1.3

[34] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *STOC*, pages 167–176, 2011. 1

[35] Y. Meng Chee, T. Feng, S. Ling, H. Wang, and L. F. Zhang. Query-Efficient Locally Decodable Codes of Subexponential Length. *ArXiv e-prints*, August 2010. 1.1, 1.3, 2.2.3, 4.2, 4.2.3

[36] Jean Pierre. Serre. *Linear representations of finite groups / Jean-Pierre Serre ; translated from the French by Leonard L. Scott*. Springer-Verlag, New York :, 1977. 3.1.1

[37] Madhu Sudan. *Efficient Checking of Polynomials and Proofs anf the Hardness of Approximation Problems*. PhD thesis, University of California at Berkeley, 1992. (document)

[38] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. 1.3, 1.3

[39] Luca Trevisan. List Decoding Using the XOR Lemma. In *FOCS*, pages 126–135, 2003. 1.3

[40] Luca Trevisan. Some applications of coding theory in computational complexity. Technical Report 043, Electronic Colloquium on Computational Complexity (ECCC), 2004. 1, 2.1.3

[41] David Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 2007. 1

[42] David P. Woodruff. Corruption and recovery-efficient locally decodable codes. In *APPROX-RANDOM*, pages 584–595, 2008. 1.3

[43] David P. Woodruff and Sergey Yekhanin. A geometric approach to information-theoretic private information retrieval. *SIAM J. Comput.*, 37(4):1046–1056, 2007. (document)

[44] Wozencraft. list decoding. *Quart. Progr. Rep., Res. Lab. Electron*, 1958. 1.3

[45] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008. (document), 1, 1.1, 2.2

[46] Sergey Yekhanin. Locally decodable codes. *"Foundations and trends in the- oretical computer science"*, 2010. 1