Foundations and Trends$^{\circledR}$ in
sample
Vol. xx, No xx (xxxx) 1–86
© xxxx xxxxxxxxx
DOI: xxxxxx

Locally Decodable Codes

Sergey Yekhanin[1]

[1] *Microsoft Research Silicon Valley, 1065 La Avenida, Mountain View, CA, 94043, USA, yekhanin@microsoft.com*

Abstract

Locally decodable codes are a class of error-correcting codes. Error-correcting codes help ensure reliable transmission of information over noisy channels. Such codes allow one to add redundancy, or bit strings, to messages, encoding them into longer bit strings, called codewords, in a way that the message can still be recovered even if a certain fraction of the codeword bits are corrupted. In typical applications of error-correcting codes the message is first partitioned into small blocks, each of which is then encoded separately. This encoding strategy allows efficient random-access retrieval of the information, since one must decode only the portion of data in which one is interested. Unfortunately, this strategy yields poor noise resilience, since, when even a single block is completely corrupted, some information is lost. In view of this limitation it would seem preferable to encode the whole message into a single codeword of an error-correcting code. Such a solution improves the robustness to noise but is hardly satisfactory, since one needs to look at the whole codeword in order to recover any particular bit of the message.

Locally decodable codes are codes that simultaneously provide efficient random-access retrieval and high noise resilience by allowing reliable reconstruction of an arbitrary bit of the message from looking at only a small number of randomly chosen codeword bits. Local decodability comes at the price of certain loss in terms of code efficiency. Specifically, locally decodable codes require longer codeword lengths than their classical counterparts. This book introduces and motivates locally decodable codes, and discusses the central results of the subject, with the main focus on the recent constructions of codes from families of “matching” vectors.

Contents

1

Introduction

Locally Decodable Codes (LDCs) are a special kind of error-correcting codes. Error-correcting codes are used to ensure reliable transmission of information over noisy channels as well as to ensure reliable storage of information on a medium that may be partially corrupted over time (or whose reading device is subject to errors). In both of these applications the message is typically partitioned into small blocks and then each block is encoded separately. Such encoding strategy allows efficient random-access retrieval of the information, since one needs to decode only the portion of data one is interested in. Unfortunately, this strategy yields very poor noise resilience, since in case even a single block (out of possibly tens of thousands) is completely corrupted some information is lost. In view of this limitation it would seem preferable to encode the whole message into a single codeword of an error-correcting code. Such solution clearly improves the robustness to noise, but is also hardly satisfactory, since one now needs to look at the whole codeword in order to recover any particular bit of the message (at least in the case when classical error-correcting codes are used). Such decoding complexity is prohibitive for modern massive data-sets.

Locally decodable codes are error-correcting codes that avoid the problem mentioned above by having extremely efficient *sublinear-time* decoding algorithms. More formally, an r-query locally decodable code C encodes k-bit messages $\mathbf{x}$ in such a way that one can probabilistically recover any bit $\mathbf{x}(i)$ of the message by querying only r bits of the (possibly corrupted) codeword $C(\mathbf{x})$, where r can be as small as 2.

Hadamard code. The classical Hadamard code [73] encoding k-bit messages to 2^k-bit codewords provides the simplest nontrivial example of locally decodable codes. In what follows, let $[k]$ denote the set $\{1,\ldots,k\}$. Every coordinate in the Hadamard code corresponds to one (of 2^k) subsets of $[k]$ and stores the XOR of the corresponding bits of the message $\mathbf{x}$. Let $\mathbf{y}$ be an (adversarially corrupted) encoding of $\mathbf{x}$. Given an index $i \in [k]$ and $\mathbf{y}$, the Hadamard decoder picks a set S in $[k]$ uniformly at random and outputs the XOR of the two coordinates of $\mathbf{y}$ corresponding to sets S and $S \triangle \{i\}$. (Here, $\triangle$ denotes the symmetric difference of sets such as $\{1,4,5\} \triangle \{4\} = \{1,5\}$, and $\{1,4,5\} \triangle \{2\} = \{1,2,4,5\}$). It is not difficult to verify that if $\mathbf{y}$ differs from the correct encoding of $\mathbf{x}$ in at most δ fraction of coordinates than with probability $1-2\delta$ both decoder's queries go to uncorrupted locations. In such case, the decoder correctly recovers the i-th bit of $\mathbf{x}$. The Hadamard code allows for a super-fast recovery of the message bits (such as, given a codeword corrupted in 0.1 fraction of coordinates, one is able to recover any bit of the message with probability 0.8 by reading only two codeword bits).

The main parameters of interest in locally decodable codes are the codeword length and the query complexity. The length of the code measures the amount of redundancy that is introduced into the message by the encoder. The query complexity counts the number of bits that need to be read from the (corrupted) codeword in order to recover a single bit of the message. Ideally, one would like to have both of these parameters as small as possible. One however can not minimize the length and the query complexity simultaneously. There is a trade-off. On one end of the spectrum we have classical error correcting codes [73, 95] that have both query complexity and codeword length proportional to the message length. On the other end we have the Hadamard code that has query complexity 2 and codeword length exponential in the message length. Establishing the optimal trade-off between the length and the query complexity is the major goal of research in the area of locally decodable codes.

Interestingly, the natural application of locally decodable codes to data transmission and storage described above is neither the historically earliest nor the most important. LDCs have a host of applications in other areas including cryptography [27, 58], complexity theory [93, 35], data structures [28, 26], derandomization [37], and the theory of fault tolerant computation [84].

1.1 The history of locally decodable codes

Locally decodable codes can be seen as the combinatorial analogs of self-correctors [70, 21] that have been studied in complexity theory in the late 1980s. LDCs were also explicitly discussed in the PCP literature in early 1990s, most notably in [6, 88, 80]. However the first formal definition of LDCs was given only in 2000 by Katz and Trevisan [64]. See also Sudan et al. [90]. Since then the study of LDCs has grown into a fairly broad field.

1.1.1 Constructions

One can informally classify the known families of locally decodable codes into three generations based on the technical ideas that underlie them. The first generation [64, 13] captures codes that are based on the idea of polynomial interpolation. Messages are encoded by complete evaluations of low degree multivariate polynomials over a finite field. Local decodability is achieved through reliance on the rich structure of short local dependencies between such evaluations at multiple points. The ideas behind the first generation of locally decodable codes go back to classical codes [73, 95], named after their discoverers, Reed and Muller. Muller discovered the codes [74] in the 1950s, and Reed proposed the majority logic decoding [83]. In what follows we often refer to locally decodable codes of the first generation as Reed-Muller (RM) LDCs. The method behind codes of the first generation is very general. In particular it yields LDCs for all possible query complexities, i.e., one can choose r to be an arbitrary function of the message length

k. For constant query complexity $r \geq 2$, RM LDCs have codeword length $\exp\left(k^{1/(r-1)}\right)$.[1]

The second generation of locally decodable codes [14, 100] started with a breakthrough paper of Beimel et al. [14] from 2002 that combined the earlier ideas of polynomial interpolation with a clever use of recursion to show that (contrary to an earlier conjecture from [27]) Reed-Muller type codes are not optimal. The code construction of [14] is indirect. Firstly, one obtains certain cryptographic protocols called Private Information Retrieval schemes, or PIRs, that on their own, are objects of interest. Secondly, one turns PIRs into LDCs. Locally decodable codes of the second generation that are capable of tolerating a constant fraction of errors, are known to exist only for constant values of r, i.e., values of r that are independent of the message length k. The codeword length is given by $\exp\left(k^{O\left(\frac{\log\log r}{r\log r}\right)}\right)$.

The latest (third) generation of locally decodable codes [101, 81, 65, 38, 61, 36, 17] was initiated by the author of this survey [101] in 2006. New codes are obtained through an argument involving a mixture of combinatorial and algebraic ideas, where the key ingredient is a design of a large family of low-dimensional (matching) vectors with constrained dot products. In what follows we often refer to locally decodable codes of the third generation as Matching Vector (MV) locally decodable codes. We summarize the most important developments related to MV codes below.

- The first family of MV codes was obtained in [101]. There are two concepts central to the construction, namely, combinatorial and algebraic niceness of subsets of finite fields. Combinatorial niceness measures the size of the family of matching vectors that underlies the code. Algebraic niceness indicates whether it is possible to reduce the query complexity of an MV code below the value one gets from the basic construction.

  The construction in [101] relied on a family of matching vectors modulo a prime. It gave 3-query LDCs of length

[1] Throughout the book we use the standard notation $\exp(x) = 2^{O(x)}$.

$\exp\left(k^{O(1/\log\log k)}\right)$, under the assumption that there exist infinitely many Mersenne primes, and 3-query codes of length $\exp\left(k^{10^{-7}}\right)$ unconditionally (casting a strong doubt in earlier conjectures [44, 50] about the length of optimal locally decodable codes).

- Raghavendra [81] suggested an alternative "homomorphism-centric" view of the construction from [101] that later turned out very important.
- Kedlaya et al. [65] argued that one cannot considerably improve parameters of MV codes from [101] by generalizing the construction to work starting from families of matching vectors over arbitrary (not necessarily prime) fields.
- Building on Raghavendra's view of MV codes Efremenko [38] generalized the code construction to work starting from families of matching vectors modulo composites. Efremenko [38] relied on a powerful combinatorial construction of matching vectors due to Grolmusz [53, 54] and for every positive integer $t \geq 2$, obtained a family of 2^t-query LDCs of length $\exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t}\right)$. The construction relied only on the concept of combinatorial niceness (but not on algebraic niceness).

  Efremenko [38] also obtained the first family of 3-query locally decodable codes that unconditionally have subexponential length. That construction relied on both algebraic and combinatorial niceness.
- Itoh and Suzuki [61] showed that in certain cases the query complexity of codes from [38] can be reduced, and obtained the shortest currently known families of locally decodable codes with constant query complexity.
- Dvir et al. [36] introduced yet another "polynomial-centric" view of MV codes and studied code parameters in the regime of super-constant query complexity. They proved that MV locally decodable codes are superior to LDCs of earlier generations for query complexities smaller than $\log k/(\log\log k)^{O(1)}$, and that MV codes are inferior to LDCs

of the first generation for query complexities larger than $(\log k)^{\Omega(\sqrt{\log k})}$.

- Ben-Aroya et al. [17] independently rediscovered some of the results of [36]. They also showed that MV codes can be made to tolerate the optimal fraction of errors. (That is, $1/2 - \epsilon$ fraction of errors over large alphabets, and $1/4 - \epsilon$ over the binary alphabet). Finally, they studied local list-decoding of MV codes.

In a related work Woodruff [98] obtained a number of results relating the query complexity and the codeword length of locally decodable codes to the fraction of noise that they can tolerate.

1.1.2 Lower bounds

Existing lower bounds for the length of locally decodable codes fit the following high level strategy. Firstly, one converts a given locally decodable code, into a *smooth* code, i.e., a code where each query of the decoder is distributed (nearly) uniformly over the set of codeword coordinates. Secondly, one employs either classical combinatorial tools such as isoperimetric inequalities and random restrictions [64, 51, 30, 78] or quantum information theory inequalities [66, 96, 97] to obtain a bound on the length of the smooth code.

The first lower bounds for the length of locally decodable codes were obtained by Katz and Trevisan [64]. Further work on lower bounds includes [51, 30, 78, 66, 96, 97, 99]. It is known that 1-query LDCs do not exist [64]. The length of optimal 2-query LDCs was settled by Kerenidis and de Wolf in [66] and is exponential in the message length. However for values of query complexity $r \geq 3$ we are still very far from closing the gap between lower and upper bounds. Specifically, the best lower bounds to date are of the form $\tilde{\Omega}\left(k^{1+1/(\lceil r/2 \rceil - 1)}\right)$ due to Woodruff [97], while the best upper bounds are $\exp\exp\left((\log k)^{O(1/\log r)}(\log\log k)^{1-\Omega(1/\log r)}\right)$ [38, 61, 36].

Interestingly, in a recent paper Gal and Mills [43] show that 3-query LDCs, correcting more than 1/3 fraction of errors require exponential length.

1.2 Organization

The goal of this survey is to summarize the state of the art in locally decodable codes. Our main focus is on codes arising from families of matching vectors. An earlier survey of LDC literature has been written by Trevisan [93] in 2004.

Our book is organized into seven chapters. In chapter 2 we formally define locally decodable codes and give a detailed treatment of Reed-Muller LDCs. Chapter 3 is dedicated to a detailed review of matching vector codes. We present a transformation that turns an arbitrary family of matching vectors into a family of locally decodable codes. We provide a detailed comparison between the parameters of MV LDCs (based on the currently largest known matching families [53]) and RM LDCs. Our presentation mostly follows [36]. We also cover some results from [17]. Chapter 4 contains a systematic study of families of matching vectors. We present the family of matching vectors due to Grolmusz as well a few other families. We also discuss upper bounds on the size of matching families. Section 4.3 of that chapter contains previously unpublished work. The rest of the chapter is based on [53, 54] and [36]. In chapter 5 we deal with lower bounds for the length of locally decodable codes and cover results from [64, 66].

In chapter 6 we discuss three most prominent applications of locally decodable codes, namely, applications to private information retrieval schemes [27], secure multi party computation [58], and circuit lower bounds [35]. Finally, in the last chapter we list (and comment on) the most exciting open questions relating to locally decodable codes and private information retrieval schemes.

2

Preliminaries

In this chapter we formally define locally decodable and locally correctable codes, and study parameters of locally decodable codes of the first generation. We start by setting up the notation and terminology used in the remainder of the book.

- $[k] = \{1, \ldots, k\}$;
- $\mathbb{F}_q$ is a finite field of q elements;
- $\mathbb{F}_q^*$ is the multiplicative group of $\mathbb{F}_q$;
- $(\mathbf{x}, \mathbf{y})$ stands for the dot product of vectors $\mathbf{x}$ and $\mathbf{y}$;
- $d(\mathbf{x}, \mathbf{y})$ denotes the Hamming distance between vectors $\mathbf{x}$ and $\mathbf{y}$, i.e., the number of coordinates where $\mathbf{x}$ and $\mathbf{y}$ differ;
- For a vector $\mathbf{w} \in \mathbb{F}_q^n$ and an integer $l \in [n]$, $\mathbf{w}(l)$ denotes the l-th coordinate of $\mathbf{w}$;
- A D-evaluation of a function h defined over a domain D, is a vector of values of h at all points of D;
- With a slight abuse of terminology we often refer to a dimension n of a vector $\mathbf{x} \in \mathbb{F}_q^n$ as its *length*.

We now proceed to define locally decodable codes.

2.1 Locally decodable and locally correctable codes

A q-ary LDC encoding k-long messages to N-long codewords has three parameters: r, δ, and ϵ. Informally an (r, δ, ϵ)-locally decodable code encodes k-long messages $\mathbf{x}$ to N-long codewords $C(\mathbf{x})$, such that for every $i \in [k]$, the coordinate value $\mathbf{x}_i$ can be recovered with probability $1 - \epsilon$, by a randomized decoding procedure that makes only r queries, even if the codeword $C(x)$ is corrupted in up to δN locations. Formally,

Definition 2.1. A q-ary code $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$ is said to be (r, δ, ϵ)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that

(1) For all $\mathbf{x} \in \mathbb{F}_q^k$, $i \in [k]$ and all vectors $\mathbf{y} \in \mathbb{F}_q^N$ such that $d(C(\mathbf{x}), \mathbf{y}) \leq \delta N$:

$$\Pr[\mathcal{A}^{\mathbf{y}}(i) = \mathbf{x}(i)] \geq 1 - \epsilon,$$

where the probability is taken over the random coin tosses of the algorithm $\mathcal{A}$.

(2) $\mathcal{A}$ makes at most r queries to $\mathbf{y}$.

We would like to have LDCs that for a given message length k and alphabet size q have small values of r, N and ϵ and a large value of δ. However typically the parameters are not regarded as equally important. In applications of locally decodable codes to data transmission and storage one wants δ to be a large constant, (ideally, close to $1/4$ for binary codes), and the codeword length N to be small. At the same time the exact number of queries r is not very important provided that it is much smaller than k. Similarly the exact value of $\epsilon < 1/2$ is not important since one can easily amplify ϵ to be close to 0, by running the decoding procedure few times and taking a majority vote. At the same time in applications of locally decodable codes in cryptography one thinks of $\delta > 0$ and $\epsilon < 1/2$ as constants whose values are of low significance and focuses on the trade-off between r and N, with emphasis on very small values of r such as $r = 3$ or $r = 4$.

A locally decodable code is called *linear* if C is a linear transformation over $\mathbb{F}_q$. All constructions of locally decodable codes considered in the book yield linear codes. While our main interest is in binary codes we deal with codes over larger alphabets as well.

A locally decodable code allows to probabilistically decode any coordinate of a message by probing only few coordinates of its corrupted encoding. A stronger property that is desirable in certain applications is that of local correctability [70, 20, 9], allowing to efficiently recover not only coordinates of the message but also arbitrary coordinates of the encoding. Locally decodable codes of the first generation that we discuss in the next sections are locally correctable.

Definition 2.2. A code (set) C in the space $\mathbb{F}_q^N$ is (r, δ, ϵ)-locally correctable if there exists a randomized correcting algorithm $\mathcal{A}$ such that

(1) For all $\mathbf{c} \in C$, $i \in [N]$ and all vectors $\mathbf{y} \in \mathbb{F}_q^N$ such that $d(\mathbf{c}, \mathbf{y}) \leq \delta N$:

$$\Pr[\mathcal{A}^{\mathbf{y}}(i) = \mathbf{c}(i)] \geq 1 - \epsilon,$$

where the probability is taken over the random coin tosses of the algorithm $\mathcal{A}$.

(2) $\mathcal{A}$ makes at most r queries to $\mathbf{y}$.

The next lemma shows how one can obtain a locally decodable code from any locally correctable code that is a linear subspace of $\mathbb{F}_q^N$. Using Sauer lemma [62] one can prove an analogous statement for general (i.e., non-linear) locally correctable codes.

Lemma 2.3. Let q be a prime power. Suppose $C \subseteq \mathbb{F}_q^N$ is a (r, δ, ϵ)-locally correctable code that is linear subspace; then there exists a q-ary (r, δ, ϵ)-locally decodable linear code C' encoding messages of length $\dim C$ to codewords of length N.

Proof. Let $I \subseteq [N]$ be a set of $k = \dim C$ information coordinates of C, (i.e., a set of coordinates whose values uniquely determine an element of C.) For $\mathbf{c} \in C$ let $\mathbf{c}|_I \in \mathbb{F}_q^k$ denote the restriction of $\mathbf{c}$ to coordinates in I. Given a message $\mathbf{x} \in \mathbb{F}_q^k$ we define $C'(\mathbf{x})$ to be the unique element

$\mathbf{c} \in C$ such that $c|_I = \mathbf{x}$. It is easy to see that local correctability of C yields local decodability of C'. □

In what follows we often implicitly assume the translation between locally correctable linear codes and locally decodable codes. Specifically, we sometimes talk about message length (rather than dimension) of such codes.

2.2 Reed-Muller locally decodable codes

A Reed Muller locally decodable code [73, 95] is specified by three integer parameters. Namely, a prime power (alphabet size) q, number of variables n, and a degree $d < q - 1$. The q-ary code consists of $\mathbb{F}_q^n$-evaluations of all polynomials of total degree at most d in the ring $\mathbb{F}_q[z_1, \ldots, z_n]$. Such code encodes $k = \binom{n+d}{d}$-long messages over $\mathbb{F}_q$ to q^n-long codewords. In sections 2.2.1–2.2.3 we consider three local correctors (decoders) for RM codes of increasing level of sophistication. Finally, in section 2.2.4 we show how one can turn non-binary RM LDCs into binary.

2.2.1 Basic decoding on lines

In this section we present the simplest local corrector for Reed Muller codes [11, 70]. To recover the value of a degree d polynomial $F \in \mathbb{F}_q[z_1, \ldots, z_n]$ at a point $\mathbf{w} \in \mathbb{F}_q^n$ it shoots a random affine line through $\mathbf{w}$ and then relies on the local dependency between the values of F at some $d + 1$ points along the line.

Proposition 2.4. Let n and d be positive integers. Let q be a prime power, $d < q-1$; then there exists a linear code of dimension $k = \binom{n+d}{d}$ in $\mathbb{F}_q^N$, $N = q^n$, that is $(d + 1, \delta, (d + 1)\delta)$-locally correctable for all δ.

Proof. The code consists of $\mathbb{F}_q^n$-evaluations of all polynomials of total degree at most d in the ring $\mathbb{F}_q[z_1, \ldots, z_n]$. The local correction procedure is the following. Given an evaluation of a polynomial F corrupted in up to δ fraction of coordinates and a point $\mathbf{w} \in \mathbb{F}_q^n$ the local corrector picks a vector $\mathbf{v} \in \mathbb{F}_q^n$ uniformly at random and considers a line

$$L = \{\mathbf{w} + \lambda\mathbf{v} \mid \lambda \in \mathbb{F}_q\}$$

through $\mathbf{w}$. Let S be an arbitrary subset of $\mathbb{F}_q^*$, $|S| = d+1$. The corrector queries coordinates of the evaluation vector corresponding to points $\mathbf{w} + \lambda\mathbf{v}$, $\lambda \in S$ to obtain values $\{e_\lambda\}$. Next, it recovers the unique univariate polynomial h, $\deg h \leq d$, such that $h(\lambda) = e_\lambda$, for all $\lambda \in S$, and outputs $h(0)$.

Note that in case all queries of our corrector go to uncorrupted locations h is the restriction of F to L, and $h(0) = F(\mathbf{w})$. It remans to note that since each individual query of the corrector goes to a uniformly random location, with probability at least $1 - (d+1)\delta$, it never query a corrupted coordinate. □

We say that an r-query code C *tolerates* a δ fraction of errors if C is (r, δ, ϵ)-locally correctable (decodable) for some $\epsilon < 1/2$. Observe that codes given by proposition 2.4 can only tolerate $\delta < 1/2(d+1)$. Thus the fraction tolerable noise rapidly deteriorates with an increase in the query complexity. In the following section we present a better local corrector for RM codes that tolerates δ close to $1/4$ independent of the number of queries.

2.2.2 Improved decoding on lines

The local corrector presented below goes back to Gemmell et al. [45]. In contrast to the setting of proposition 2.4 we require that d is substantially smaller than q. To recover the value of a degree d polynomial $F \in \mathbb{F}_q[z_1, \ldots, z_n]$ at a point $\mathbf{w} \in \mathbb{F}_q^n$ the corrector shoots a random affine line through $\mathbf{w}$ and then relies on the high redundancy among the values of F along the line.

Proposition 2.5. Let $\sigma < 1$ be a positive real. Let n and d be positive integers. Let q be a prime power such that $d < \sigma(q-1)$; then there exists a linear code of dimension $k = \binom{n+d}{d}$ in $\mathbb{F}_q^N$, $N = q^n$, that is $(q-1, \delta, 2\delta/(1-\sigma))$-locally correctable for all δ.

Proof. The code is exactly the same as in proposition 2.4, and the correction procedure is related to the procedure above. Given a δ-corrupted evaluation of a degree d polynomial F and a point $\mathbf{w} \in \mathbb{F}_q^n$ the corrector picks a vector $\mathbf{v} \in \mathbb{F}_q^n$ uniformly at random and considers a line

$$L = \{\mathbf{w} + \lambda \mathbf{v} \mid \lambda \in \mathbb{F}_q\}$$

through $\mathbf{w}$. The corrector queries coordinates of the evaluation vector corresponding to points $\mathbf{w} + \lambda \mathbf{v}$, $\lambda \in \mathbb{F}_q^*$ to obtain values $\{e_\lambda\}$. Next, it recovers the unique univariate polynomial h, $\deg h \leq d$, such that $h(\lambda) = e_\lambda$, for all but at most $\lfloor (1-\sigma)(q-1)/2 \rfloor$ values of $\lambda \in \mathbb{F}_q^*$, and outputs $h(0)$. If such a polynomial h does not exist the corrector outputs 0. The search for h can be done efficiently using the Berlekamp-Welch algorithm [73] for decoding Reed Solomon codes.

It remans to note that since each individual query of the corrector goes to a uniformly random location, by Markov's inequality the probability that more than $\lfloor (1-\sigma)(q-1)/2 \rfloor$ of the queries go to corrupted locations is at most $2\delta/(1-\sigma)$. Therefore with probability at least $1 - 2\delta/(1-\sigma)$, h is the restriction of F to L, and $h(0) = F(\mathbf{w})$. □

When σ is small the local corrector given by proposition 2.5 tolerates a nearly 1/4 fraction of errors. In the following section we present an even better corrector that tolerates a nearly 1/2 fraction of errors, which is optimal for unique decoding.

2.2.3 Decoding on curves

The local corrector presented below goes back to Gemmell and Sudan [46]. Again we require that d is substantially smaller than q. To recover the value of a degree d polynomial $F \in \mathbb{F}_q[z_1, \ldots, z_n]$ at a point $\mathbf{w} \in \mathbb{F}_q^n$ the corrector shoots a random parametric degree two curve through $\mathbf{w}$ and then relies on the high redundancy among the values of F along the curve. The advantage upon the decoder of proposition 2.5 comes from the fact that points on a random curve (in contrast to points on a random line) constitute a two-independent sample from the underlying space.

Proposition 2.6. Let $\sigma < 1$ be a positive real. Let n and d be positive integers. Let q be a prime power such that $d < \sigma(q-1)$; then there exists a linear code of dimension $k = \binom{n+d}{d}$ in $\mathbb{F}_q^N$, $N = q^n$, that for all positive $\delta < 1/2 - \sigma$ is $(q-1, \delta, O_{\sigma,\delta}(1/q))$-locally correctable.

Proof. The code is exactly the same as in propositions 2.4 and 2.5, and the correction procedure is related to the procedures above. Given a δ-corrupted evaluation of a degree d polynomial F and a point $\mathbf{w} \in \mathbb{F}_q^n$ the corrector picks vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$ uniformly at random and considers a degree two curve

$$\chi = \{\mathbf{w} + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2 \mid \lambda \in \mathbb{F}_q\}$$

through $\mathbf{w}$. The corrector tries to reconstruct a restriction of F to χ, which is a polynomial of degree up to $2d$. To this end the corrector queries coordinates of the evaluation vector corresponding to points $\chi(\lambda) = \mathbf{w} + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2$, $\lambda \in \mathbb{F}_q^*$ to obtain values $\{e_\lambda\}$. Next, it recovers the unique univariate polynomial h, $\deg h \leq 2d$, such that $h(\lambda) = e_\lambda$, for all but at most $\lfloor (1-2\sigma)(q-1)/2 \rfloor$ values of $\lambda \in \mathbb{F}_q^*$, and outputs $h(0)$. If such a polynomial h does not exist the corrector outputs 0. It is not hard to verify that the corrector succeeds if the number of queries that go to corrupted locations is at most $\lfloor (1-2\sigma)(q-1)/2 \rfloor$.

Below we analyze the success probability of the corrector. For $\mathbf{a} \in \mathbb{F}_q^n$ and $\lambda \in \mathbb{F}_q^*$ consider a random variable $x_{\mathbf{a}}^\lambda$, which is the indicator variable of the event $\chi(\lambda) = \mathbf{a}$. Let $E \subseteq \mathbb{F}_q^n$, $|E| \leq \delta N$ be the set of $\mathbf{a} \in \mathbb{F}_q^n$ such that the values of F at $\mathbf{a}$ are corrupted. For every $\lambda \in \mathbb{F}_q^*$ consider a random variable

$$x^\lambda = \sum_{\mathbf{a} \in E} x_{\mathbf{a}}^\lambda.$$

Note that variables $\left\{x^\lambda\right\}, \lambda \in \mathbb{F}_q^*$ are pairwise independent. For every $\lambda \in \mathbb{F}_q^*$ we have

$$\mathbb{E}\left[x^\lambda\right] \leq \delta \quad \text{and} \quad \mathbb{D}\left[x^\lambda\right] \leq \delta - \delta^2.$$

Finally consider a random variable

$$x = \sum_{\lambda \in \mathbb{F}_q^*} x^\lambda,$$

that counts the number of corrector's queries that go to corrupted locations. By pairwise independence we have

$$\mathbb{E}[x] \leq (q-1)\delta \text{ and } \mathbb{D}[x] \leq (q-1)(\delta - \delta^2).$$

By Chebychev's inequality [3] we have

$$\Pr\left[\mathrm{x} \geq \left\lfloor \frac{(1-2\sigma)(\mathrm{q}-1)}{2} \right\rfloor\right] \leq \frac{4(\delta-\delta^2)}{(\mathrm{q}-1)(1-2(\sigma+\delta))^2} = \mathrm{O}_{\sigma,\delta}\left(\frac{1}{\mathrm{q}}\right).$$

This concludes the proof. □

2.2.4 Binary codes

Propositions 2.4, 2.5, and 2.6 yield non-binary codes. As we stated earlier our main interest is in binary codes. The next lemma extends proposition 2.6 to produce binary codes that tolerate a nearly 1/4 fraction of errors, which is optimal for unique decoding over $\mathbb{F}_2$. The idea behind the proof is fairly standard and involves concatenation [41, 73].

Proposition 2.7. Let $\sigma < 1$ be a positive real. Let n and d be positive integers. Let $q = 2^b$ be a power of two such that $d < \sigma(q-1)$. Suppose that there exists a binary linear code C_{inner} of distance μB encoding b-bit messages to B-bit codewords; then there exists a linear code C of dimension $k = \binom{n+d}{d} \cdot b$ in $\mathbb{F}_2^N$, $N = q^n \cdot B$, that for all positive $\delta < (1/2 - \sigma)\mu$ is $((q-1)B, \delta, O_{\sigma,\mu,\delta}(1/q))$-locally correctable.

Proof. We define the code C to be the concatenation [73, 95, 41] of the q-ary code C_{outer} used in propositions 2.4–2.6 and the binary code C_{inner}. In order to recover a single bit, the local corrector recovers the symbol of the q-ary alphabet that the bit falls into. Given a δ-corrupted concatenated evaluation of a degree d polynomial F and a point $\mathbf{w} \in \mathbb{F}_q^n$ the corrector acts similarly to the corrector from the proposition 2.6. Specifically, it picks vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$ uniformly at random and considers a degree two curve

$$\chi = \{\mathbf{w} + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2 \mid \lambda \in \mathbb{F}_q\}$$

through $\mathbf{w}$. To recover $F(\mathbf{w})$ the corrector attempts to reconstruct a restriction of F to χ, which is a polynomial of degree up to $2d$.

To this end the corrector queries all $(q-1)B$ codeword coordinates corresponding to encodings of values of F at points $\chi(\lambda) = \mathbf{w} + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2$, $\lambda \in \mathbb{F}_q^*$ and then recovers the unique univariate polynomial $h \in \mathbb{F}_q[\lambda]$, $\deg h \leq 2d$, such that C_{inner}-encodings of values of h along $\mathbb{F}_q^*$ agree with all but at most $\lfloor (1-2\sigma)\mu(q-1)B/2 \rfloor$ observed binary values. If such a polynomial h does not exist the corrector outputs 0. It is not hard to verify that the corrector succeeds if the number of queries that go to corrupted locations is at most $\lfloor (1-2\sigma)\mu(q-1)B/2 \rfloor$. Decoding can be done efficiently provided that C_{inner} has an efficient decoder [40, 41].

Below we analyze the success probability of the corrector. For every $\mathbf{a} \in \mathbb{F}_q^n$ let $t_{\mathbf{a}}$ denote the number of corrupted coordinates in the C_{inner}-encoding of the value of F at $\mathbf{a}$. We have

$$\sum_{\mathbf{a} \in \mathbb{F}_q^n} t_{\mathbf{a}} \leq \delta q^n B.$$

For $\mathbf{a} \in \mathbb{F}_q^n$ and $\lambda \in \mathbb{F}_q^*$ consider a random variable $x_{\mathbf{a}}^{\lambda}$, which is the indicator variable of the event $\chi(\lambda) = \mathbf{a}$. For every $\lambda \in \mathbb{F}_q^*$ consider a random variable

$$x^{\lambda} = \sum_{\mathbf{a} \in \mathbb{F}_q^n} t_{\mathbf{a}} x_{\mathbf{a}}^{\lambda}.$$

Note that variables $\left\{x^{\lambda}\right\}, \lambda \in \mathbb{F}_q^*$ are pairwise independent. For every $\lambda \in \mathbb{F}_q^*$ we have

$$\mathbb{E}\left[x^{\lambda}\right] \leq \delta B \quad \text{and} \quad \mathbb{D}\left[x^{\lambda}\right] \leq (\delta - \delta^2) B^2.$$

Finally consider a random variable

$$x = \sum_{\lambda \in \mathbb{F}_q^*} x^{\lambda},$$

that counts the number of corrector's queries that go to corrupted locations. By pairwise independence we have

$$\mathbb{E}[x] \leq \delta(q-1)B \quad \text{and} \quad \mathbb{D}[x] \leq (\delta - \delta^2)(q-1)B^2.$$

By Chebychev's inequality [3] we have

$$\Pr\left[\mathrm{x} \geq \left\lfloor \frac{(1-2\sigma)\mu(\mathrm{q}-1)\mathrm{B}}{2} \right\rfloor\right] \leq$$
$$\leq \frac{4(\delta-\delta^2)}{(q-1)((1-2\sigma)\mu-2\delta)^2} = O_{\sigma,\mu,\delta}\left(\frac{1}{q}\right).$$

This concludes the proof. □

Propositions 2.6 and 2.7 give locally correctable codes that tolerate the amount of error that is nearly optimal for unique (even non-local) decoding ($1/2$ fraction of errors over large alphabets, $1/4$ over $\mathbb{F}_2$). An important model of error correction that generalizes unique decoding is that of list decoding [39, 55]. In that model the decoder is allowed to output a small list of codewords rather than a single codeword. Decoding is considered successful if the transmitted codeword appears in the list. List decoding allows for error-correction beyond the "half the minimum distance barrier". One can show that Reed Muller codes are *locally* list decodable from the nearly optimal amount of noise [5, 90] ($1-\epsilon$ fraction of errors over large alphabets, $1/2-\epsilon$ over $\mathbb{F}_2$). However we are not going to discuss these results in this book.

2.3 Summary of parameters

In the previous section we gave a detailed treatment of locally decodable codes of the first generation. These codes enjoy a few remarkable properties. Specifically, they yield the shortest known LDCs when the query complexity is sufficiently large ($r \geq \log k/(\log\log k)^c$, for some constant c) and they constitute *the only* known family of locally correctable codes.

The method behind Reed Muller codes is simple and general. It yields codes for all possible values of query complexity r, i.e., one can set r to be an arbitrary function of the message length k by specifying an appropriate relation between n and d in propositions 2.5–2.7 and letting these parameters grow to infinity. Increasing d relative to n yields shorter codes of larger query complexity.

Below we present asymptotic parameters of several families of binary LDCs of the first generation.

- $r = O(1)$. Proposition 2.4 yields r-query LDCs of length $\exp\left(k^{1/(r-1)}\right)$ over an alphabet of size $O(r)$. One can get binary LDCs of the same query complexity and the same asymptotic length from private information retrieval schemes of [4, 59, 14].
- $r = O(\log k \log\log k)$. In proposition 2.7 set $d = n$, $q = cd$ for a large constant c, and let n grow while concatenating with asymptotically good binary codes of relative distance μ close to half. This yields a family of r-query binary locally correctable codes that encode k-bit messages to $k^{O(\log\log k)}$-bit codewords and tolerate a nearly $1/4$ fraction of errors (depending on c and μ).
- $r \leq (\log k)^t$, for constant $t > 1$. In proposition 2.7 set $d = n^t$, $q = cd$ and let n grow while concatenating with asymptotically good binary codes of relative distance close to half. This yields a family of r-query binary locally correctable codes that encode k-bit messages to $k^{1+1/(t-1)+o(1)}$-bit codewords and tolerate a nearly $1/4$ fraction of errors.
- $r = O(k^{1/t} \log k)$, for integer constant $t \geq 1$. In proposition 2.7 set $n = t$, $q = cd$ and let d grow while concatenating with asymptotically good binary codes of relative distance close to half. This yields a family of r-query binary locally correctable codes that encode k-bit messages to $t^{t+o(t)} \cdot k$-bit codewords and tolerate a nearly $1/4$ fraction of errors.

We summarize the parameters of binary locally correctable codes obtained above in the following table.

r	N
$O(1)$	$\exp\left(k^{1/(r-1)}\right)$
$O(\log k \log\log k)$	$k^{O(\log\log k)}$
$(\log k)^t, t > 1$	$k^{1+1/(t-1)+o(1)}$
$O(k^{1/t} \log k), t \geq 1$	$t^{t+o(t)} \cdot k$

3

Matching vector codes

In this chapter we give a detailed treatment of locally decodable codes that arise from families of matching vectors. Any construction of such codes naturally falls into two parts: the design of a matching vector family, and the actual code construction. Here we focus on the second part and defer an in-depth study of matching vector families to chapter 4.

The chapter is organized into seven sections. In section 3.1 we explain the intuition behind matching vector codes and setup the language that is used later. Our presentation follows the latest "polynomial-centric" view of MV codes that fleshes out some intrinsic similarity between MV codes and Reed Muller codes. In sections 3.2–3.4 we discuss three local decoders for matching vector codes of increasing level of sophistication. In section 3.5 we show how one can turn non-binary matching vector codes into binary. Finally, in sections 3.6 and 3.7 we summarize asymptotic parameters of MV codes and provide a detailed comparison between matching vector locally decodable codes and Reed Muller locally decodable codes.

3.1 The framework

Our constructions are centered around a "polynomial-centric" view of MV codes [36] that fleshes out some intrinsic similarity between matching vector codes and Reed Muller codes. In this view an MV code consists of a linear subspace of polynomials in $\mathbb{F}_q[z_1, \ldots, z_n]$, evaluated at all points of $\mathbb{C}_m^n$, where $\mathbb{C}_m$ is a certain multiplicative subgroup of $\mathbb{F}_q^*$. The decoding algorithm is similar to traditional local decoders for RM codes given by propositions 2.4–2.5. The decoder shoots a line in a certain direction and decodes along it. The difference is that the monomials which are used are not of low-degree, they are chosen according to a matching family of vectors. Further, the lines for decoding are *multiplicative*, a notion that we define shortly. In what follows let $\mathbb{Z}_m$ denote the ring of integers modulo an integer m.

Definition 3.1. Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. We say that families $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ and $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of vectors in $\mathbb{Z}_m^n$ form an S-matching family if the following two conditions are satisfied:

- For all $i \in [k]$, $(\mathbf{u}_i, \mathbf{v}_i) = 0$;
- For all $i, j \in [k]$ such that $i \neq j$, $(\mathbf{u}_j, \mathbf{v}_i) \in S$.

We now show how one can obtain an matching vector locally decodable code out of a matching family. We start with some notation.

- We assume that q is a prime power, m divides $q - 1$, and denote the unique subgroup of $\mathbb{F}_q^*$ of order m by $\mathbb{C}_m$;
- We fix some generator g of $\mathbb{C}_m$;
- For $\mathbf{w} \in \mathbb{Z}_m^n$, we define $g^{\mathbf{w}} \in \mathbb{C}_m^n$ by $\left(g^{\mathbf{w}(1)}, \ldots, g^{\mathbf{w}(n)}\right)$;
- For $\mathbf{w}, \mathbf{v} \in \mathbb{Z}_m^n$ we define the multiplicative line $M_{\mathbf{w},\mathbf{v}}$ through $\mathbf{w}$ in direction $\mathbf{v}$ to be the multi-set
$$M_{\mathbf{w},\mathbf{v}} = \left\{ g^{\mathbf{w}+\lambda\mathbf{v}} \mid \lambda \in \mathbb{Z}_m \right\}; \tag{3.1}$$
- For $\mathbf{u} \in \mathbb{Z}_m^n$, we define the monomial $\mathrm{mon}_{\mathbf{u}} \in \mathrm{\mathbb{F}_q[z_1, \ldots, z_n]}$ by
$$\mathrm{mon}_{\mathbf{u}}(\mathrm{z}_1, \ldots, \mathrm{z}_n) = \prod_{\ell \in [\mathrm{n}]} \mathrm{z}_\ell^{\mathbf{u}(\ell)}. \tag{3.2}$$

Observe that for any $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbb{Z}_m^n$ and $\lambda \in \mathbb{Z}_m$ we have

$$\mathrm{mon}_{\mathbf{u}} \left(g^{\mathbf{w}+\lambda\mathbf{v}} \right) = g^{(\mathbf{u},\mathbf{w})} \left(g^{\lambda} \right)^{(\mathbf{u},\mathbf{v})}. \tag{3.3}$$

The formula above implies that the $M_{\mathbf{w},\mathbf{v}}$-evaluation of a monomial $\mathrm{mon}_{\mathbf{u}}$ is a $\mathbb{C}_m$-evaluation of a (univariate) monomial

$$g^{(\mathbf{u},\mathbf{w})} y^{(\mathbf{u},\mathbf{v})} \in \mathbb{F}_q[y]. \tag{3.4}$$

This observation is the foundation of our decoding algorithms. We now specify the encoding procedure and outline the main steps taken by all decoding procedures described later on (propositions 3.2, 3.4, 3.5, and 3.6). Let $\mathcal{U}, \mathcal{V}$ be an S-matching family in $\mathbb{Z}_m^n$.

Encoding: We encode a message $(\mathbf{x}(1), \ldots, \mathbf{x}(k)) \in \mathbb{F}_q^k$ by the $\mathbb{C}_m^n$-evaluation of the polynomial

$$F(z_1, \ldots, z_n) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot \mathrm{mon}_{\mathbf{u}_\mathrm{j}}(\mathrm{z}_1, \ldots, \mathrm{z}_\mathrm{n}). \tag{3.5}$$

Notice that $F = F_{\mathbf{x}}$ is a function of the message $\mathbf{x}$ (we will omit the subscript and treat $\mathbf{x}$ as fixed throughout this section).

Abstract decoding: The input to the decoder is a corrupted $\mathbb{C}_m^n$-evaluation of F and an index $i \in [k]$.

(1) The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ uniformly at random;
(2) The decoder recovers the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$. To accomplish this the decoder may query the corrupted $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at m or fewer locations.

To see that noiseless $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F uniquely determines $\mathbf{x}(i)$ note that by formulas (3.3), (3.4) and (3.5) the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is a $\mathbb{C}_m$-evaluation of a polynomial

$$f(y) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot g^{(\mathbf{u}_j,\mathbf{w})} y^{(\mathbf{u}_j,\mathbf{v}_i)} \in \mathbb{F}_q[y]. \tag{3.6}$$

Further observe that the properties of the S-matching family $\mathcal{U}, \mathcal{V}$ and (3.6) yield

$$f(y) = \mathbf{x}(i) \cdot g^{(\mathbf{u}_i,\mathbf{w})} + \sum_{s \in S} \left(\sum_{j \,:\, (\mathbf{u}_j,\mathbf{v}_i)=s} \mathbf{x}(j) \cdot g^{(\mathbf{u}_j,\mathbf{w})} \right) y^s. \tag{3.7}$$

For a polynomial $h \in \mathbb{F}_q[y]$ we denote by supp(h) the set of monomials with non zero coefficients in h, where a monomial y^e is identified with the integer e. It is evident from formula (3.7) that supp(f) $\subseteq$ S $\cup$ $\{0\}$ and

$$\mathbf{x}(i) = f(0)/g^{(\mathbf{u}_i,\mathbf{w})}. \tag{3.8}$$

In sections 3.2–3.4 we describe several local decoders that follow the general paradigm outlined above.

3.2 Basic decoding on lines

The proposition below gives the simplest local decoder for MV codes. In the current form it has first appeared in [38]. Earlier versions based on matching vectors modulo primes can be found in [101, 81].

Proposition 3.2. Let $\mathcal{U}, \mathcal{V}$ be a family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$, $|S| = s$. Suppose $m \mid q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long messages to m^n-long codewords that is $(s+1, \delta, (s+1)\delta)$-locally decodable for all δ.

Proof. The encoding procedure has already been specified by formula (3.5). To recover the value $\mathbf{x}(i)$

(1) The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ at random, and queries the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at $(s+1)$ consecutive locations $\left\{g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \{0, \ldots, s\}\right\}$ to obtain values $c_0, \ldots, c_s$.

(2) The decoder recovers the unique sparse univariate polynomial $h(y) \in \mathbb{F}_q[y]$ with supp(h) $\subseteq S \cup \{0\}$ such that for all $\lambda \in \{0, \ldots, s\}$, $h(g^\lambda) = c_\lambda$. (The uniqueness of $h(y)$ follows from standard properties of Vandermonde matrices. [68])

(3) Following the formula (3.8) the decoder returns $h(0)/g^{(\mathbf{u}_i,\mathbf{w})}$.

The discussion in section 3.1 implies that if all $(s + 1)$ locations queried by the decoder are not corrupted then $h(y)$ is indeed the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$, and decoder's output is correct. It remains to note that each individual query of the decoder goes to a uniformly random location and apply the union bound. □

In the proposition above we interpolate the polynomial $h(y)$ to recover its free coefficient. In certain cases (relying on special properties of the integer m and the set S) it may be possible to recover the free coefficient in ways that do not require complete interpolation and thus save on the number of queries. This general idea has been used in [101] under the name of "algebraic niceness", in [38] for the case of three-query codes, and in [61] to obtain the shortest currently know LDCs in the regime of $r = O(1)$. Currently the quantitative improvements one gets through the use of "algebraic niceness" are relatively small. Therefore we are not go into detail on them in this book.

3.3 Improved decoding on lines

The local decoder for MV codes given in section 3.2 is similar to the local decoder for RM codes given in section 2.2.1 in that it can only tolerate $\delta < 1/2r$ fraction of errors. Thus the fraction of tolerable noise rapidly deteriorates with an increase in query complexity r. Below we introduce the concept of a bounded matching family of vectors and show how matching vector codes based on bounded matching families can be decoded from a nearly $1/4$ fraction of errors independent of r.

In what follows we identify $\mathbb{Z}_m$ with the subset $\{0, \ldots, m-1\}$ of real numbers. This imposes a total ordering on $\mathbb{Z}_m$, $0 < 1 < \ldots < m-1$ and allows us to compare elements of $\mathbb{Z}_m$ with reals. We say that a set $S \subseteq \mathbb{Z}_m$ is b-bounded if for all $s \in S$, $s < b$.

Definition 3.3. Let b be a positive real. An S-matching family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_m^n$ is b-bounded if the set S is b-bounded.

The proposition below is due to Dvir et al. [36]. Some ideas behind it were independently rediscovered by Ben-Aroya et al. [17].

Proposition 3.4. Let σ be a positive real. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$. Suppose $m \mid q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long messages to m^n-long codewords that is $(m, \delta, 2\delta/(1-\sigma))$-locally decodable for all δ.

Proof. The encoding procedure has already been specified by (3.5). To recover the value $\mathbf{x}(i)$,

(1) The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ at random, and queries every point of the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at all m locations $\left\{g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \mathbb{Z}_m\right\}$ to obtain values $c_0, \ldots, c_{m-1}$.
(2) The decoder recovers the univariate polynomial $h(y) \in \mathbb{F}_q[y]$ of degree less than σm such that for all but at most $(m - \sigma m)/2$ values of $\lambda \in \mathbb{Z}_m$, $h(g^\lambda) = c_\lambda$. If such an h does not exist the decoder encounters a failure, and returns 0. Note that $\deg h < \sigma m$ implies that $h(y)$ is unique, if it exists. The search for $h(y)$ can be done efficiently using the Berlekamp-Welch algorithm [73].
(3) Following the formula (3.8) the decoder returns $h(0)/g^{(\mathbf{u}_i,\mathbf{w})}$.

The discussion in section 3.1 implies that if the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is corrupted in at most $(m - \sigma m)/2$ locations, then $h(y)$ is indeed the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$, and the decoder's output is correct. It remains to note that each individual query of the decoder goes to a uniformly random location and thus by Markov's inequality the probability that more than $(m - \sigma m)/2$ of decoder's queries go to corrupted locations is at most $2\delta/(1 - \sigma)$. □

3.3.1 Further improvement for small S

The local decoder of proposition 3.4 does not use any information about the size of the set S (only the fact that all elements in S are bounded). Below we show how one can reduce the query complexity in the cases when $|S|$ is small and $\ln q$ is small relative to m. [36]

Proposition 3.5. Let σ be a positive real. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$, $|S| = s$. Suppose $m \mid q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long messages to m^n-long codewords that is (r, δ, ϵ)-locally decodable for all $0 < \alpha < 1 - \sigma, \delta$, where

$$r = \left\lceil (s+2) \ln q/\alpha^2 \right\rceil, \qquad \epsilon = 2\delta/(1 - \sigma - \alpha).$$

Proof. Our decoding algorithm is similar to the one of proposition 3.4. The saving in the number of queries comes from the fact that the decoder does not query all points on the multiplicative line but rather partitions the line into classes, and queries all points within a certain class. Our proof consists of two parts. Firstly, we establish the existence of an appropriate partition. Secondly, we present the decoding algorithm. We start with some notation. Let $\alpha > 0$ be fixed.

- Let $L \subseteq \mathbb{F}_q[y]$ be the linear space of polynomials whose support is contained in $\{0\} \cup S$;
- Let $T \subseteq \mathbb{Z}_m$. We say that T is α-regular, if for all $h \in L$ we have
$$\left| T \cap \left\{ \lambda \in \mathbb{Z}_m \mid h(g^\lambda) = 0 \right\} \right| < (\sigma + \alpha)|T|; \tag{3.9}$$
- Let $t \leq m$ be a fixed positive integer. Let π be a partition of $\mathbb{Z}_m$ into $p = \lfloor m/t \rfloor$ classes where each class is of size t or more
$$\mathbb{Z}_m = \bigsqcup_{\ell=1}^{p} \pi_\ell; \tag{3.10}$$
- We say that π is α-regular, if for each $\ell \in [p]$, π_ℓ, is α-regular.

We now argue that for a sufficiently large t, there exists a partition π satisfying (3.10) that is α-regular. Fix an arbitrary non-zero polynomial $h \in L$. Let $W = \left\{ \lambda \in \mathbb{Z}_m \mid h(g^\lambda) = 0 \right\}$. Clearly, $|W| < \sigma m$. Fix $t' \geq t$ and pick a set $T \subseteq \mathbb{Z}_m$ of size exactly t' uniformly at random.

$$\begin{aligned} \Pr\left[|T \cap W| \geq (\sigma + \alpha)t'\right] = \Pr\left[|T \cap W| - \sigma t' \geq \alpha t'\right] \leq \\ \Pr\left[|T \cap W| - \mathbb{E}(|T \cap W|) > \alpha t'\right] \leq \exp(-2\alpha^2 t), \end{aligned} \tag{3.11}$$

where the last inequality follows from [34, theorem 5.3].

Now let $t = \lceil (s+2) \ln q / 2\alpha^2 \rceil$. If $t > m$; then the proposition trivially follows from the proposition 3.4. We assume $t \leq m$ and pick π to be a random partition satisfying (3.10). Clearly, no class in π has size more than $2t - 1$. Relying on (3.11), the union bound, and $m/t < q$ we conclude that π is α-regular with positive probability since

$$(m/t)(q^{(s+1)} - 1) < e^{2\alpha^2 t}. \tag{3.12}$$

Fix an α-regular partition π. We are now ready to define the code. The encoding procedure has already been specified by formula (3.5). To recover the value $\mathbf{x}(i)$

(1) The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ and $\ell \in [p]$ uniformly at random, and queries points of the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at $|\pi_\ell|$ locations $\left\{g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \pi_\ell\right\}$ to obtain values $\{c_\lambda \mid \lambda \in \pi_\ell\}$.

(2) The decoder recovers the univariate polynomial $h(y) \in \mathbb{F}_q[y]$ with $\operatorname{supp}(h) \subseteq \{0\} \cup S$ such that for all but at most $(1-\sigma-\alpha)|\pi_\ell|/2$ values of $\lambda \in \pi_\ell$, $h(g^\lambda) = c_\lambda$. If such an h does not exist the decoder encounters a failure, and returns 0. Note that the properties of π imply that $h(y)$ is unique, if it exists.

(3) Following the formula (3.8) the decoder returns $h(0)/g^{(\mathbf{u}_i,\mathbf{w})}$.

The discussion in section 3.1 implies that if at most $(1-\sigma-\alpha)|\pi_l|/2$ locations queried by the decoder are corrupted; then $h(y)$ is indeed the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$, and the decoder's output is correct. It remains to note that each individual query of the decoder goes to a uniformly random location and thus by Markov's inequality the probability that more than $(1-\sigma-\alpha)|\pi_l|/2$ queries go to corrupted locations is at most $2\delta/(1-\sigma-\alpha)$, and to observe that the total number of queries is at most $2t-1$. □

3.4 Decoding on collections of lines

The local decoder for MV codes given in section 3.3 is similar to the local decoder for RM codes given in section 2.2.2 in that it can only tolerate $\delta < 1/4$ fraction of errors. Below we present an even better local decoder that tolerates a nearly $1/2$ fraction of errors, which is optimal for unique decoding. All results in this section are due to Ben-Aroya et al. [17].

The idea behind the improved local decoder is different from the idea behind the Gemmell Sudan decoder [46] for Reed Muller codes (proposition 2.6). There we exploited restrictions of RM codewords to parametric degree two curves. It is however not clear how to utilize similar restrictions in the setting of matching vector codes. Instead, we randomly pick a sufficiently large (but constant) number of multiplicative lines. We assign weights to candidate values of the desired message symbol based on the number of errors along the collection of lines. We argue that the symbol with the largest weight is with high probability the correct symbol. This technique bears some similarity to Forney's GMD decoding [40, 41].

Proposition 3.6. Let σ be a positive real. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$. Suppose $m \mid q-1$, where q is a prime power; then for every positive integer l there exists a q-ary linear code encoding k-long messages to m^n-long codewords that for all $\delta < (1-\sigma)/2$ is $(lm, \delta, \exp_{\sigma,\delta}(-l))$-locally decodable.

Proof. The encoding procedure has already been specified by (3.5). We setup the notation needed to describe the decoder. Given a polynomial $h(y) \in \mathbb{F}_q[y]$, $\mathrm{supp(h)} \subseteq \mathrm{S} \cup \{0\}$ and a multiplicative line M we denote the number of coordinates where $\mathbb{C}_m$-evaluation of h agrees with the M-evaluation of F by $\mathrm{agr(h, M)}$. For a symbol $e \in \mathbb{F}_q$ and a multiplicative line M we define

$$\mathrm{weight(e, M)} = \max_{\mathrm{h:h(0)=e}} \mathrm{agr(h, M)},$$

where the maximum is taken over all $h(y) \in \mathbb{F}_q[y]$, $\mathrm{supp(h)} \subseteq \mathrm{S} \cup \{0\}$. We now proceed to the local decoder. To recover the value $\mathbf{x}(i)$,

(1) The decoder picks vectors $\mathbf{w}_1, \ldots, \mathbf{w}_l \in \mathbb{Z}_m^n$ uniformly at random, and queries values of the corrupted evaluation of F along each of l multiplicative lines $\left\{M_{\mathbf{w}_j, \mathbf{v}_i}\right\}$, $j \in [l]$.

(2) For every symbol $e \in \mathbb{F}_q$ the decoder computes its weight,

$$\mathrm{weight(e)} = \sum_{\mathrm{j=1}}^{\mathrm{l}} \mathrm{weight}\left(\mathrm{e}, \mathrm{M}_{\mathbf{w}_\mathrm{j}, \mathbf{v}_\mathrm{i}}\right).$$

The weight measures the likelihood that $\mathbf{x}(i) = e$ given the observed values of the corrupted evaluation of F.

(3) The decoder outputs the symbol that has the largest weight. If such a symbol is not unique the decoder outputs 0.

Below we analyze the success probability of the decoder. Firstly, note that there cannot be two symbols $e_1 \neq e_2$ that both have weight above $lm(1+\sigma)/2$. Otherwise one of the multiplicative lines would give us two distinct polynomials $h_1(y), h_2(y) \in \mathbb{F}_q[y]$ of degree less than σm whose $\mathbb{C}_m$-evaluations agree in at least σm locations. Secondly, note that by Chernoff bound [34] the probability that the total number of corrupted locations on lines $\left\{M_{\mathbf{w}_j,\mathbf{v}_i}\right\}$, $j \in [l]$ exceeds $lm(1-\sigma)/2$ is at most $\exp_{\sigma,\delta}(-l)$, provided that $\delta < (1-\sigma)/2$. □

3.5 Binary codes

In cases when query complexity r is super-constant propositions 3.2, 3.4, 3.5 and 3.6 yield codes over growing alphabets. As we stated earlier our main interest is in binary codes. The next lemma extends proposition 3.6 to produce binary codes that tolerate a nearly 1/4 fraction of errors, which is optimal for unique decoding over $\mathbb{F}_2$. The proof uses standard concatenation [41, 73].

Proposition 3.7. Let σ be a positive real. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$. Suppose $m \mid q-1$, where $q = 2^b$. Further suppose that there exists a binary linear code C_{inner} of distance μB encoding b-bit messages to B-bit codewords; then for every positive integer l there exists a binary linear code C encoding kb-bit messages to $m^n \cdot B$-bit codewords that for all $\delta < (1-\sigma)\mu/2$ is $(lmB, \delta, \exp_{\sigma,\mu,\delta}(-l))$-locally decodable.

Proof. We define the code C to be the concatenation [73, 95, 41] of the q-ary code C_{outer} used in propositions 3.2–3.6 and the binary code C_{inner}. In order to recover a single bit, the local decoder recovers the symbol of the q-ary alphabet that the bit falls into. Given a δ-corrupted concatenated evaluation of a polynomial F the decoder acts similarly to the decoder from the proposition 3.6.

We setup the notation needed to describe the decoder formally. Given a polynomial $h(y) \in \mathbb{F}_q[y]$, supp(h) $\subseteq$ S$\cup\{0\}$ and a multiplicative line M we denote the number of coordinates where C_{inner}-concatenated $\mathbb{C}_m$-evaluation of h agrees with corrupted C_{inner}-concatenated M-

evaluation of F by agr(h, M). For a symbol $e \in \mathbb{F}_q$ and a multiplicative line M we define

$$\mathrm{weight(e, M)} = \max_{\mathrm{h:h(0)=e}} \mathrm{agr(h, M)},$$

where the maximum is taken over all $h(y) \in \mathbb{F}_q[y]$, $\mathrm{supp(h)} \subseteq \mathrm{S} \cup \{0\}$. To recover the i-th symbol of the outer code,

(1) The decoder picks vectors $\mathbf{w}_1, \ldots, \mathbf{w}_l \in \mathbb{Z}_m^n$ uniformly at random, and queries the coordinates corresponding to encodings of values of F along each of l lines $\left\{M_{\mathbf{w}_j, \mathbf{v}_i}\right\}$, $j \in [l]$.

(2) For every symbol $e \in \mathbb{F}_q$ the decoder computes its weight,

$$\mathrm{weight(e)} = \sum_{\mathrm{j=1}}^{\mathrm{l}} \mathrm{weight}\left(\mathrm{e}, \mathrm{M}_{\mathbf{w}_\mathrm{j}, \mathbf{v}_\mathrm{i}}\right).$$

The weight measures the likelihood that the i-the symbol of the outer code equals e given the observed values of the corrupted evaluation of F.

(3) The decoder outputs the required bit of the symbol that has the largest weight. If such a symbol is not unique the decoder outputs 0.

Below we analyze the success probability of the decoder. Firstly, note that there cannot be two symbols $e_1 \neq e_2$ that both have weight above $lmB(1 - (1 - \sigma)\mu/2)$. Otherwise one of the multiplicative lines would give us two distinct polynomials $h_1(y), h_2(y) \in \mathbb{F}_q[y]$ of degree less than σm whose concatenated $\mathbb{C}_m$-evaluations agree in at least $(1 - (1 - \sigma)\mu)mB$ locations. Secondly, note that by Chernoff bound [34] the probability that the total number of corrupted locations on lines $\left\{M_{\mathbf{w}_j, \mathbf{v}_i}\right\}$, $j \in [l]$ exceeds $lmB(1 - \sigma)\mu/2$ is at most $\exp_{\sigma,\mu,\delta}(-l)$, provided that $\delta < (1 - \sigma)\mu/2$. □

Propositions 3.6 and 3.7 give matching vector codes that tolerate the amount of error that is nearly optimal for unique (even non-local) decoding (1/2 fraction of errors over large alphabets, 1/4 over $\mathbb{F}_2$). In [17] Ben-Aroya et al. design local decoders for MV codes that correct the nearly optimal amount of noise in the list decoding model [5, 90] ($1-\epsilon$ fraction of errors over large alphabets, $1/2-\epsilon$ over $\mathbb{F}_2$). We are not going to discuss these results in this book.

3.6 Summary of parameters

Parameters of matching vector codes (propositions 3.2–3.7) are determined by parameters of the underlying family of matching vectors. In section 3.6.1 we apply proposition 3.7 to Grolmusz's family of matching vectors to obtain some explicit trade-offs between query complexity and codeword length of MV codes. In section 3.6.2 we use existing upper bounds on the size of matching vector families to establish lower bounds on the codeword length of MV codes.

3.6.1 Upper bounds

The largest currently know families of matching vectors are closely based on Grolmusz's construction of set systems with restricted intersections modulo composites [53, 54]. The following lemma captures the parameters of these families. We defer the proof to chapter 4.

Lemma 4.8. Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Let $\{e_i\}$, $i \in [t]$ be integers such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Then there exists an $\binom{h}{w}$-sized σm-bounded family of matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$ and σ is an arbitrary real larger than $\sum_{i\in[t]} 1/p_i$.

A combination of proposition 3.7 and lemma 4.8 yields

Lemma 3.8. Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Suppose integers $\{e_i\}$, $i \in [t]$ are such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Let σ be an arbitrary real number larger than $\sum_{i\in[t]} 1/p_i$. Suppose $m \mid q-1$, where $q = 2^b$. Further suppose that there exists a binary linear code C_{inner} of distance μB encoding b-bit messages to B-bit codewords; then for every positive integer l there exists a binary linear code C encoding $\binom{h}{w} \cdot b$-bit messages to $m^{\binom{h}{\leq d}} \cdot B$-bit codewords that for all $\delta < (1-\sigma)\mu/2$ is $(lmB, \delta, \exp_{\sigma,\mu,\delta}(-l))$-locally decodable.

In what follows we estimate asymptotic parameters of our codes.

Lemma 3.9. For all integers $t \geq 2$, $k \geq 2^t$ there exists a binary linear code encoding k-bit messages to

$$N = \exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t} t \ln t\right)$$

-bit codewords that is $(r, \delta, \exp(-t))$-locally decodable for $r = t^{O(t)}$ and $\delta = 1/4 - O(1/\ln t)$.

Proof. The proof follows by setting parameters in lemma 3.8.

(1) By [87, theorem 5.7] there exists a universal constant c' such that the range $[(c'/2)t\ln t, c't\ln t]$ contains at least t distinct odd primes $p_1, \ldots, p_t$;
(2) Note that $\sum_{i\in[t]} 1/p_i = O(1/\ln t)$;
(3) Set $m = \prod_{i\in[t]} p_i$. Clearly, $m = t^{\Theta(t)}$;
(4) Set b to be the smallest positive integer such that $m \,\big|\, 2^b - 1$ Clearly, $b = t^{O(t)}$. Set $q = 2^b$;
(5) A standard greedy argument (that is used to prove the classical Gilbert-Varshamov bound [73, 95]) implies that there is a universal constant c'' such that for all integers $s \geq 1$, there exists a binary linear code of distance $(1/2 - c''/\sqrt{s})s^2$ encoding s-bit messages to s^2-bit codewords. We set C_{inner} to be a binary linear code that encodes b-bit messages to $B = t^{\Theta(t)}$-bit codewords and has distance μB, for $\mu \geq \left(1/2 - c''/t^{\Theta(t)}\right)$;
(6) We now assume that there exists a positive integer w which is a multiple of t such that $k = w^{w/t}$. Clearly, we have $w = \Theta(t\log k/\log\log k)$;
(7) Following lemma 3.8 for every $i \in [t]$, let e_i be the smallest integer such that $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$. Clearly, $d = O(w^{1/t} t\ln t)$;

(8) Set $h = \lceil w^{1+1/t} \rceil$;

(9) Observe that $\binom{h}{w} \cdot b \geq (h/w)^w \geq k$;

(10) Note that $\binom{h}{\leq d} \leq d(eh/d)^d$;

(11) Set $N = m^{\binom{h}{\leq d}} \cdot B \leq t^x$, where $x = O(t)(ew)^{O(w^{1/t} t \ln t)}$;

(12) Set $l = t$;

(13) We combine lemma 3.8 with inequalities that we proved above and make basic manipulations to obtain a binary linear code encoding k-bit messages to

$$\exp\exp\left((\log k)^{1/t} (\log\log k)^{1-1/t} t \ln t\right)$$

-bit codewords that is $(r, \delta, \exp(-t))$-locally decodable for $r = t^{O(t)}$ and $\delta = 1/4 - O(1/\ln t)$;

(14) Finally, we note that the assumption about $k = w^{w/t}$, for some w can be safely dropped. If k does not have the required shape, we pad k-bit messages with zeros to get messages of length k', where k' has the shape $w^{w/t}$ and then apply the procedure above. One can easily check that such padding requires at most a quadratic blow up in the message length and therefore does not affect asymptotic parameters.

This completes the proof. □

The following theorem gives asymptotic parameters of matching vector codes in terms of the query complexity and the message length.

Theorem 3.10. For every large enough integer r and every $k \geq r$ there exists a binary linear r-query locally decodable code encoding k-bit messages to

$$\exp\exp\left((\log k)^{O(\log\log r/\log r)} (\log\log k)^{1-\Omega(\log\log r/\log r)} \log r\right) \quad (3.13)$$

bit codewords and tolerating $\delta = 1/4 - O(1/\ln\ln r)$ fraction of errors.

Proof. The proof follows by setting parameters in lemma 3.9. Set t to be the largest integer such that $t^{O(t)} \leq r$, where the constant in O-notation is the same as the one in lemma 3.9. Assuming r is sufficiently large we have $t = \Theta(\log r / \log\log r)$. One can also check that $k \geq r$ implies that the condition of lemma 3.9 is satisfied. An application of the lemma concludes the proof. □

Theorem 3.10 presents a trade-off between the query complexity and the codeword length of matching vector codes that tolerate a nearly optimal fraction of errors. In section 2.1 we mentioned that not all applications of LDCs require codes of such a high error tolerance. Specifically, applications of locally decodable codes in cryptography need short codes of constant query complexity $r = O(1)$, that tolerate some constant fraction of errors that is low significance.

It is possible to get a small saving in terms of codeword length in theorem 3.10 if one disregards the fraction of tolerable noise. Below we give two theorems that apply in that setting. The next theorem is due to Itoh and Suzuki [61]. We omit the proof that slightly improves on what one gets by a simple combination of proposition 3.2 and lemma 4.8.

Theorem 3.11. For every integer $t \geq 2$, and for all $k \geq 2$, there exists an $r = 3 \cdot 2^{t-2}$-query linear locally decodable code over $\mathbb{F}_{2^t}$ encoding k-long messages to

$$\exp\exp_t\left((\log k)^{1/t}(\log\log k)^{1-1/t}\right)$$

-long codewords and tolerating $\delta = O(1/r)$ fraction of errors.

Theorem 3.11 yields non-binary codes. One can turn these codes into binary without an increase in the number of queries using the technique from [38, section 4]. Again we omit the proof.

Theorem 3.12. For every integer $t \geq 2$, and for all $k \geq 2$, there exists a $r = 3 \cdot 2^{t-2}$-query binary linear locally decodable code encoding k-bit messages to

$$\exp\exp_t\left((\log k)^{1/t}(\log\log k)^{1-1/t}\right)$$

-bit codewords and tolerating $\delta = O(1/r)$ fraction of errors.

Locally decodable codes given by propositions 3.2–3.7 and theorems 3.10 and 3.11 are *perfectly smooth*, i.e., on an arbitrary input each individual query of the decoder is distributed perfectly uniformly over the codeword coordinates. Binary codes given by theorem 3.12 are not perfectly smooth.

To conclude we remark here that the entire construction and analysis of matching vector codes described in the preceding sections (apart from the parts dealing with reduction to the binary case) work also if the underlying field, $\mathbb{F}_q$, is replaced with the complex number field $\mathbb{C}$. The only property we used in $\mathbb{F}_q$ is that it contains an element of order m, which trivially holds over $\mathbb{C}$ for every m. This implies the existence of linear locally decodable codes with essentially the same parameters as above also over the complex numbers (the definition of locally decodable codes over an arbitrary field is the same as for finite fields, we simply allow the decoder to preform field arithmetic operations on its inputs). Once one has a linear code over the complex numbers, it is straightforward to get a code over the reals by writing each complex number as a pair of real numbers. Interestingly, other than matching vector codes (and trivial 2-query codes of exponential stretch), there are no known constructions of locally decodable codes neither over $\mathbb{C}$ nor over $\mathbb{R}$. LDCs over characteristic zero have applications in arithmetic circuit complexity [37, 35].

3.6.2 Lower bounds

Let $k(m,n)$ denote the size of the largest family of S-matching vectors in $\mathbb{Z}_m^n$ where we allow S to be an arbitrary subset of $\mathbb{Z}_m \setminus \{0\}$. It is easy to see that the rate of any locally decodable code obtained via propositions 3.2– 3.7 is at most $k(m,n)/m^n$.

In this section we use existing upper bounds on the size of matching vector families to establish lower bounds on the codeword length of matching vector codes. The codeword length lower bounds we get are very general. In particular they apply to *all* matching vector codes, irrespective of their query complexity.

The following upper bounds on $k(m,n)$ are due to Dvir et al. [36]. We defer the proofs to chapter 4.

Theorem 4.23. Let m and n be arbitrary positive integers. Suppose p is a prime divisor of m; then

$$k(m,n) \leq \frac{5m^n}{p^{(n-1)/2}}.$$

Theorem 4.25. Let m and n be arbitrary positive integers; then

$$k(m,n) \leq m^{n-1+o_m(1)}.$$

We now translate upper bounds on matching vector families to lower bounds on the encoding length of matching vector codes. We first argue that any family of non-binary matching vector codes, i.e., codes that for some m and n, encode $k(m,n)$-long messages to m^n-long codewords has an encoding blow-up of at least $2^{\Omega\left(\sqrt{\log k}\right)}$.

Theorem 3.13. Consider an infinite family of matching vector codes $C_\ell : \mathbb{F}_q^k \to \mathbb{F}_q^N$ for $\ell \in \mathbb{N}$, where $k = k(\ell)$ and $N = N(\ell)$ go to infinity with ℓ. For large enough ℓ, we have

$$k \leq \frac{N}{2^{\Omega\left(\sqrt{\log N}\right)}} \quad \Rightarrow \quad N \geq k2^{\Omega\left(\sqrt{\log k}\right)}.$$

Proof. For each ℓ, we have a family of matching vectors in $\mathbb{Z}_m^n$ where m, n depend on ℓ. We have $N = m^n$ while $k \leq k(m,n)$. First assume that $n > \sqrt{\log N}$. Then by theorem 4.23 with p a prime divisor of m, we have

$$k \leq \frac{5m^n}{p^{(n-1)/2}} \leq \frac{5N}{2^{0.5\sqrt{\log N}-1/2}} \leq \frac{N}{2^{0.4\sqrt{\log N}}},$$

where the last inequality holds for large enough N, and hence for all large ℓ. Hence assume that $n \leq \sqrt{\log N}$ so that $m \geq 2^{\sqrt{\log N}}$. As ℓ

goes to infinity, N and hence m go to infinity. So for large enough ℓ, theorem 4.25 gives $k(m,n) \leq m^{n-1+o_m(1)} \leq m^{n-0.9}$. Hence

$$k \leq \frac{m^n}{m^{0.9}} \leq \frac{N}{2^{0.9\sqrt{\log N}}}.$$

Thus $k \leq \frac{N}{2^{\Omega(\sqrt{\log N})}}$ for large enough ℓ. This implies that $N \geq k 2^{\Omega\left(\sqrt{\log k}\right)}$ for large enough ℓ. □

One can generalize theorem 3.13 to get a similar statement for binary matching vector codes, i.e., codes obtained by a concatenation of a non-binary MV code with an asymptotically good binary code.

Theorem 3.14. Let $\{m_\ell\}$ and $\{n_\ell\}$, $\ell \in \mathbb{N}$ be two arbitrary sequences of positive integers, such that ${m_\ell}^{n_\ell}$ monotonically grows to infinity. Consider an infinite family of binary codes $C_\ell : \mathbb{F}_2^{k_\ell} \to \mathbb{F}_2^{N_\ell}$ for $\ell \in \mathbb{N}$, where each code C_ℓ is obtained via a concatenation of an MV code encoding $k(m_\ell, n_\ell)$-long messages to $m_\ell^{n_\ell}$-long codewords over $\mathbb{F}_{q_\ell}$, (here $q_\ell = 2^t$ is the smallest such that $m_\ell \mid 2^t - 1$) with an asymptotically good binary code of some fixed rate; then for large enough ℓ the relative redundancy of C_ℓ is at least $2^{\Omega\left(\sqrt{\log k_\ell}\right)}$.

Proof. Pick a sufficiently large value of ℓ. Consider two cases

- $n_\ell \neq 1$. It is not hard to see that $k(m_\ell, n_\ell) \geq k(m_\ell, 2) \geq m_\ell$. Now note that by theorem 3.13 relative redundancy of the non-binary code is at least $2^{\Omega\left(\sqrt{\log k(m_\ell,n_\ell)}\right)}$, and the concatenation with a binary code can only increase relative redundancy. Finally note that the dimension k_ℓ of the binary code is at most $k(m_\ell, n_\ell) \cdot m_\ell \leq k^2(m_\ell, n_\ell)$. Thus

$$2^{\Omega\left(\sqrt{\log k(m_\ell,n_\ell)}\right)} \geq 2^{\Omega\left(\sqrt{\log k_\ell}\right)},$$

  for an appropriately chosen constant in Ω notation.
- $n_\ell = 1$. Set $k' = k(m_\ell, n_\ell)$. Be theorem 4.25, $k' = m_\ell^{o(1)}$. Note that $k_\ell = k't$ and $N_\ell = \Omega(m_\ell \cdot t)$, for some $t \leq m_\ell$. These conditions yield $N_\ell \geq \Omega\left(k_\ell^{3/2}\right)$.

This completes the proof. □

3.7 MV codes vs. RM codes

In this section we provide a comparison between matching vector codes and Reed Muller codes. We show that matching vector codes given by theorems 3.10, 3.12 have shorter codeword lengths than Reed Muller codes when the query complexity is low,

$$r \leq \log k/(\log\log k)^c,$$

for some constant c. We also show that all matching vector codes have longer codeword lengths than Reed Muller codes when the query complexity is high,

$$r \geq (\log k)^{c(\sqrt{\log k})},$$

for some constant c.

Recall that a Reed Muller locally decodable code (section 2.2) is specified by three integer parameters. Namely, a prime power (alphabet size) q, a number of variables n, and a degree $d < q-1$. The q-ary code consists of $\mathbb{F}_q^n$-evaluations of all polynomials in $\mathbb{F}_q[z_1, \ldots, z_n]$ of total degree at most d. Such code encodes $k = \binom{n+d}{d}$-long messages to q^n-long codewords and has query complexity $r = q-1$. If $d < \sigma(q-1)$, the code tolerates $\delta = 1/2 - \sigma$ fraction of errors. When q is a power of 2 non-binary RM LDCs can be turned into binary via concatenation. Concatenation with an asymptotically good code of relative distance μ yields an r-query binary linear code encoding k-bit messages to N-bit codewords and tolerating $\delta = (1/2 - \sigma)\mu$ fraction of errors, where

$$k = \binom{n+d}{d} \log q, \quad N = \Theta(q^n \log q), \quad r = \Theta(q \log q). \tag{3.14}$$

3.7.1 Low query complexity regime

We now argue that RM LDCs are inferior to codes of theorems 3.10, 3.12 for all $r \leq \log k/(\log\log k)^c$, where c is a universal constant. To arrive at such a conclusion we need a lower bound on the codeword length of Reed Muller locally decodable codes.

Let d, n, and q be such that formulas (3.14) yield an r-query LDC, where r belongs to the range of our interest. We necessarily have $d \leq n$

(otherwise $r > \log k$). Thus

$$k = \binom{n+d}{d} \log q \leq (en/d)^d \log q \leq n^{O(d)}, \tag{3.15}$$

and $n \geq k^{\Omega(1/d)}$. Therefore writing $\exp(x)$ to denote $2^{\Omega(x)}$, we have

$$N \geq \exp\exp\left(\log k/d\right) \geq \exp\exp\left(\log k/r\right). \tag{3.16}$$

Note that when r is a constant then already 3-query codes of theorem 3.12 improve substantially upon (3.16). To conclude the argument one needs to verify that there exists a constant c such that for every nondecreasing function $r(k)$, where $r(k)$ grows to infinity, and satisfies $r(k) \leq \log k/(\log\log k)^c$, for all sufficiently large k the right hand side of (3.16) evaluates to a larger value than (3.13).

3.7.2 High query complexity regime

Here we argue that all matching vector codes have longer codeword lengths than Reed Muller codes when $r \geq (\log k)^{c(\sqrt{\log k})}$, where c is a universal constant. Given the theorem 3.14 all we need to do is for every constant c' construct binary Reed Muller LDCs that have a blow-up of less than $2^{c'\sqrt{\log k}}$ and query complexity of $(\log k)^{O\left(\sqrt{\log k}\right)}$. By formula (3.14) the relative redundancy of any RM LDC specified by parameters n, d and q is given by

$$k/N \leq O\left(\binom{n+d}{d} / q^n\right).$$

We assume that $n < d$; then $\binom{n+d}{d} \leq (2ed/n)^n$. Therefore (relying of $d \leq q$) we get

$$k/N \leq O((2e/n)^n).$$

Thus to have relative redundancy below $2^{c'\sqrt{\log k}}$ it suffices to have

$$n = O_{c'}\left(\sqrt{\log k}/\log\log k\right). \tag{3.17}$$

Given k we choose n to be the largest integer satisfying (3.17). Next we choose d to be the smallest integer satisfying $k \leq \binom{n+d}{d} \log q$. One can easily check that this yields $d = (\log k)^{O(\sqrt{\log k})}$, giving an RM LDC with desired parameters.

4

Matching vectors

In the previous chapter we have seen how parameters of matching vector locally decodable codes are governed by the parameters of the underlying families of matching vectors. This chapter contains a systematic study of such families.

In the first three sections we deal with constructions. In section 4.1 we present a bounded family of matching vectors based on the Grolmusz's construction of set systems with restricted intersections modulo composites. This family underlies the main families of matching vector codes (theorems 3.10–3.12). In section 4.2 we present an elementary construction of a bounded family of matching vectors. This family improves upon the Grolmusz's family for large values of the modulus m. Finally, in section 4.3 we obtain an algebraic construction of an asymptotically optimal matching family in the narrow case of 4-dimensional vectors modulo a prime. This result has not been published previously.

In sections 4.4–4.6 we deal with upper bounds on the size of matching families. We gradually build up the necessary machinery and in section 4.6 prove theorems 4.23 and 4.25 that have been used in the previous chapter to establish lower bounds on the codeword length of matching vector codes (theorems 3.13 and 3.14).

4.1 The Grolmusz family

The construction of the matching family presented below is modeled along the lines of Grolmusz's construction of set systems with restricted intersections modulo composites [53, 54]. Grolmusz's original construction uses the low-degree OR representations of Barrington et al. [10]. However, we will use lemma 4.2 to bypass the set system and go directly to the matching family from polynomials. In addition to being more direct, this also gives a slightly larger collection of vectors. Our presentation follows [36]. We first show how to get a family of matching vectors that is not bounded, and then in section 4.1.1 show how to turn this family into a bounded one.

Definition 4.1. Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. We say that a set of polynomials $\mathcal{F} = \{f_1, \ldots, f_k\} \subseteq \mathbb{Z}_m[z_1, \ldots, z_h]$ and a set of points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \subseteq \mathbb{Z}_m^h$ form a polynomial S-matching family of size k if

- For all $i \in [k]$, $f_i(\mathbf{x}_i) = 0$;
- For all $i, j \in [k]$ such that $i \neq j$, $f_j(\mathbf{x}_i) \in S$.

Let $\mathcal{F}, \mathcal{X}$ be a k-sized polynomial matching family. For $i \in [k]$, let $\mathrm{supp(f_i)}$ denote the set of monomials in the support of the polynomial f_i. We define $\mathrm{supp}(\mathcal{F}) = \bigcup_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{supp(f_i)}$ and $\dim(\mathcal{F}) = |\mathrm{supp}(\mathcal{F})|$. The following lemma was observed by Sudan [89].

Lemma 4.2. An k-sized polynomial S-matching family $\mathcal{F}, \mathcal{X}$ over $\mathbb{Z}_m$ yields a k-sized S-matching family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_m^n$, where $n = \dim(\mathcal{F})$.

Proof. Let $\mathrm{mon_1}, \ldots, \mathrm{mon_n}$ be the set of monomials in $\mathrm{supp}(\mathcal{F})$. For every $j \in [k]$ we have

$$f_j(z_1 \ldots, z_h) = \sum_{l=1}^{n} c_{jl} \mathrm{mon_l}.$$

We define the vector $\mathbf{u}_j$ to be the n-dimensional vector of coefficients of the polynomial f_j. Similarly, for $i \in [k]$, we define the vector $\mathbf{v}_i$ to be the vector of evaluations of monomials $\mathrm{mon}_1, \ldots, \mathrm{mon}_\mathrm{n}$ at the point $\mathbf{x}_i$. It is easy to check that for all $i, j \in [k]$, $(\mathbf{u}_j, \mathbf{v}_i) = f_j(\mathbf{x}_i)$ and hence the sets $\mathcal{U}, \mathcal{V}$ indeed form an S-matching family. □

Definition 4.3 (Canonical set). Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. The *canonical set* in $\mathbb{Z}_m$ is the set of all non-zero s such that for every $i \in [t]$, $s \in \{0, 1\} \bmod p_i$.

Our goal now is to prove the following

Lemma 4.4. Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Let $\{e_i\}$, $i \in [t]$ be integers such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Let S be the canonical set; then there exists an $\binom{h}{w}$-sized family of S-matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$.

We assume that parameters $m, t, \{p_i\}_{i \in [t]}, \{e_i\}_{i \in [t]}, w, h$, and the set S satisfy the condition of lemma 4.4 whose proof we defer. Our proof of the lemma below follows [52, theorem 2.16].

Lemma 4.5. For every $i \in [t]$, there is an explicit multilinear polynomial $f_i(z_1, \ldots, z_h) \in \mathbb{Z}_{p_i}[z_1, \ldots, z_h]$ where $\deg(f_i) \leq p_i^{e_i} - 1$ such that for $\mathbf{x} \in \{0, 1\}^h$, we have

$$f_i(\mathbf{x}) \equiv \begin{cases} 0 \bmod p_i, & \text{if } \sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod p_i^{e_i}, \\ 1 \bmod p_i, & \text{otherwise.} \end{cases}$$

Proof. Our proof relies on the classical Lucas theorem [23, p. 28], stating that for all primes p and all integers

$$b = \sum_{j \geq 0} b_j \cdot p^j, \ 0 \leq b_j < p$$

$$s = \sum_{j \geq 0} s_j \cdot p^j, \ 0 \leq s_j < p,$$

we have

$$\binom{b}{s} \equiv \prod_j \binom{b_j}{s_j} \bmod p.$$

Let $\mathbf{x} \in \{0,1\}^h$ be an arbitrary vector of Hamming weight b. Let $(b_0, b_1, \ldots)$ be a p-ary expansion of b. Further, let ℓ be an arbitrary positive integer, and let S_{p^ℓ} be the h-variate multilinear symmetric polynomial of degree p^ℓ. By the Lucas theorem we have

$$S_{p^\ell}(\mathbf{x}) \;=\; \binom{b}{p^\ell} \;\equiv\; \binom{b_\ell}{1} \prod_{j \neq \ell} \binom{b_j}{0} \;\equiv\; b_\ell \bmod p.$$

To prove the lemma we need to write the function f_i as a polynomial of degree less than $p_i^{e_i}$. Observe that $f_i : \{0,1\}^h \to \{0,1\}$ is a symmetric function, i.e., its value stays the same under an arbitrary permutation of coordinates of an input vector $\mathbf{x}$. Moreover note that the value of $f_i(\mathbf{x})$ depends only on e_i least significant digits $b_0, \ldots, b_{e_i-1}$ of the p_i-ary expansion of the Hamming weight b of $\mathbf{x}$.

Using the fact that every function from $\mathbb{Z}_{p_i}^{e_i} \to \mathbb{Z}_{p_i}$ is computed by some polynomial, f_i can be written as a polynomial $g(b_0, \ldots, b_{e_i-1})$ over $\mathbb{Z}_{p_i}$ with the degree of each $b_j \leq p - 1$. But $S_{p^j}(\mathbf{x}) \equiv b_j \bmod p_i$. Hence the polynomial

$$g\left(S_1(\mathbf{x}), \ldots, S_{p_i^{e_i-1}}(\mathbf{x})\right) \in \mathbb{Z}_{p_i}[z_1, \ldots, z_h]$$

computes the function f on binary inputs. It is a symmetric polynomial whose degree is bounded by $\sum_{j=0}^{e_i-1} p_i^j (p_i - 1) = p_i^{e_i} - 1$. □

Corollary 4.6. There is an explicit multilinear polynomial $f(z_1, \ldots, z_h) \in \mathbb{Z}_m[z_1, \ldots, z_n]$ such that for all $\mathbf{x} \in \{0,1\}^h$, we have

$$f(\mathbf{x}) = \begin{cases} 0 \bmod m, & \text{if } \sum_{l=1}^h \mathbf{x}(l) = w, \\ s \bmod m, \text{ for s} \in \text{S}, & \text{if } \sum_{l=1}^h \mathbf{x}(l) < w, \end{cases}$$

where coordinates of $\mathbf{x}$ are summed as integers.

Proof. Define the polynomial f so that for all $i \in [t]$, $f(z_1, \ldots, z_h) \equiv f_i(z_1, \ldots, z_h) \bmod p_i$. We claim that it satisfies the above requirement. Observe that by the Chinese remainder theorem

$$f(\mathbf{x}) = 0 \bmod m \quad \text{iff} \quad \text{for all } i \in [t],\ \sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod p_i^{e_i}.$$

This is equivalent to saying that

$$\sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod \prod_i p_i^{e_i}.$$

Note that for all $i \in [t]$, $p_i^{e_i} > w^{1/t}$. Hence $m = \prod_i p_i^{e_i} > w$. Thus whenever the integer sum $\sum_{l=1}^{h} \mathbf{x}(l) < w$, we have $\sum_{l=1}^{h} \mathbf{x}(l) \not\equiv w \bmod m$, which proves the claim. □

Proof. [of lemma 4.4] For every $T \subseteq [h]$ of size w, define the polynomial f_T wherein the polynomial f from corollary 4.6, we set $z_j = 0$ for $j \notin T$ (but z_j stays untouched for $j \in T$). Define $\mathbf{x}_T \in \{0,1\}^h$ to be the indicator of the set T. Viewing vectors $\mathbf{x} \in \{0,1\}^h$ as indicator vectors $\mathbf{x}_L$ for sets $L \subseteq [h]$, it is easy to check that for all $T, L \in [h]$, $f_T(\mathbf{x}_L) = f(\mathbf{x}_{L \cap T})$. Combining this with Corollary 4.6 gives

- For all $T \subseteq [h]$, where $|T| = w$, $f_T(\mathbf{x}_T) = f(\mathbf{x}_T) \equiv 0 \bmod m$,
- For all $T \neq L \subseteq [h]$, where $|T| = |L| = w$, $f_T(\mathbf{x}_L) = f(\mathbf{x}_{L \cap T}) \in S \bmod m$,

where the second bullet follows from the observation that $|L \cap T| \leq w - 1$. Thus the set of polynomials $\mathcal{F} = \{f_T\}_{T \subseteq [h], |T| = w}$ and points $\mathcal{X} = \{\mathbf{x}_T\}_{T \subseteq [h], |T| = w}$ form a polynomial S-matching family.

It is clear that $k = |\mathcal{F}| = \binom{h}{w}$. To bound n, we note that $\deg(f) \leq d$ and f is multilinear. Thus we can take $\operatorname{supp}(\mathcal{F})$ to be the set of all multilinear monomials in variables $z_1, \ldots, z_h$ of degree at most d. Clearly, this yields $\dim(\mathcal{F}) = \binom{h}{\leq d}$. □

4.1.1 A bounded family

The following lemma shows that the canonical set can be turned into a bounded one via scaling by an invertible element. It has been observed in [36] and independently in [17]. Let $\mathbb{Z}_m^*$ denote the is the set of invertible elements of $\mathbb{Z}_m$.

Lemma 4.7. Let $m = \prod_{i=1}^t p_i$ be a product of distinct primes. Let S be the canonical set in $\mathbb{Z}_m$. There exists an $\alpha \in \mathbb{Z}_m^*$ such that the set αS is σm-bounded for any $\sigma > \sum_{i \in [t]} 1/p_i$.

Proof. We start with some notation.

- For every $i \in [t]$, define the integer $\hat{p_i} = m/p_i$;
- Let $\alpha \in \mathbb{Z}_m^*$ be the unique element such that for all $i \in [t], \alpha = \hat{p_i} \bmod p_i$.

Observe that for any $i, j \in [t]$,

$$\left(\alpha^{-1}\hat{p_i}\right) \bmod p_j = \begin{cases} 1, & \text{if i=j;} \\ 0, & \text{otherwise.} \end{cases}$$

Let $s \in S$ be arbitrary. Set $I = \{i \in [t] \mid p_i \text{ does not divide s}\}$. Observe that $s = \alpha^{-1} \sum_{i \in I} \hat{p_i}$. Therefore

$$\alpha s = \sum_{i \in I} \hat{p_i} \leq m \sum_{i \in [t]} 1/p_i.$$

This concludes the proof. □

The argument above shows that any S-matching family $\mathcal{U}, \mathcal{V}$ where S is the canonical set can be turned into a bounded one (by scaling all vectors in $\mathcal{V}$ by an invertible element). Combining lemma 4.7 with lemma 4.4 we obtain

Lemma 4.8. Let $m = \prod_{i=1}^t p_i$ be a product of distinct primes. Let w be a positive integer. Suppose integers $\{e_i\}$, $i \in [t]$ are such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Let S be the canonical set modulo m; then there exists an $\binom{h}{w}$-sized family of S-matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$.

4.2 An elementary family

In this section we give an elementary construction of a bounded family of matching vectors. The construction works for both prime and composite moduli. The family improves upon the family of lemma 4.8 for large values of m. In what follows we use $\mathbb{Z}_{\geq 0}$ to denote the set of non-negative integers.

Definition 4.9. Let $b(m', n)$ denote the number of vectors $\mathbf{w} \in \mathbb{Z}_{\geq 0}^n$ such that $\|\mathbf{w}\|_2^2 = m'$.

Thus $b(m', n)$ counts the number of integer points on the surface of the n-dimensional ball of radius $\sqrt{m'}$ in the positive orthant.

Lemma 4.10. Let $m' < m$ and $n \geq 2$ be arbitrary positive integers. There exists a $b(m', n-1)$-sized $(m'+1)$-bounded family of matching vectors in $\mathbb{Z}_m^n$.

Proof. Let $k = b(m', n-1)$ and let $\mathbf{w}_1, \ldots, \mathbf{w}_k$ be the vectors in $\mathbb{Z}_{\geq 0}^{n-1}$ such that $\|\mathbf{w}_i\|_2^2 = m'$. For each $\mathbf{w}_i$, we define vectors in $\mathbb{Z}^n$ by

$$\mathbf{u}_i = (1, -\mathbf{w}_i), \ \mathbf{v}_i = (m', -\mathbf{w}_i).$$

We claim that the resulting family of vectors is a $\{1, \ldots, m'\}$-matching family. To prove this, observe that $(\mathbf{u}_i, \mathbf{v}_j) = m' - (\mathbf{w}_i, \mathbf{w}_j)$. If $i = j$, then $(\mathbf{w}_i, \mathbf{w}_j) = \|\mathbf{w}_i\|_2^2 = m'$ whereas if $i \neq j$; then by Cauchy-Schwartz

$$(\mathbf{w}_i, \mathbf{w}_j) \leq \|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2 = m'.$$

In fact the inequality must be strict since $\mathbf{w}_i$ and $\mathbf{w}_j$ both lie on the surface of the same ball, hence they are not collinear. But since their inner product lies in $\mathbb{Z}_{\geq 0}$, we conclude that

$$(\mathbf{w}_i, \mathbf{w}_j) \in \{0, \ldots, m'-1\},$$

hence $(\mathbf{u}_i, \mathbf{v}_j) \in \{1, \ldots, m'\}$. Now note that since $m > m'$, the intersections do not change modulo m. □

The lemma below follows by combining lemma 4.10 with some crude lower bounds for $b(m', n-1)$.

Lemma 4.11. Let $m' < m$ and $n \geq 2$ be arbitrary positive integers. There exists a k-sized $(m'+1)$-bounded family of matching vectors in $\mathbb{Z}_m^n$, where

$$k = \tfrac{1}{m'+1}\left(\tfrac{m'}{n-1}\right)^{(n-1)/2} \qquad \text{for } m' \geq n, \tag{4.1}$$

$$k = \tbinom{n-1}{m'} \qquad \text{for } m' < n. \tag{4.2}$$

Proof. To prove (4.1), we set $d = \left\lfloor \sqrt{m'/(n-1)} \right\rfloor$. For every vector $\mathbf{w} \in \{0, \ldots, d\}^{n-1}$, we have $0 \leq \|\mathbf{w}\|^2 \leq (n-1)d^2 \leq m'$. By the pigeonhole principle, there exists some $m'' \in \{0, \ldots, m'\}$ such that $b(m'', n-1) \geq (d+1)^{n-1}/(m'+1)$, which by lemma 4.10 yields an $(m'+1)$-bounded matching family of size

$$k \geq \frac{1}{m'+1}\left(\left\lfloor \sqrt{\frac{m'}{n-1}} \right\rfloor + 1\right)^{n-1} \geq \frac{1}{m'+1}\left(\frac{m'}{n-1}\right)^{(n-1)/2}.$$

Note that the condition $m' \geq n$ is only needed to ensure that the bound in meaningful.

To prove (4.2), we observe that $b(m', n-1) \geq \binom{n-1}{m'}$ by taking all vectors in $\{0,1\}^{n-1}$ of Hamming weight exactly m'. The bound follows from lemma 4.10. □

It is interesting to observe that while matching vector codes of theorem 3.10 improve upon Reed Muller locally decodable codes only when $r \leq \log k/(\log\log k)^c$, one can get MV codes that asymptotically match RM LDCs of query complexity $r = \Theta(\log k \log\log k)$ combining lemma 4.11 (where m has the shape $2^b - 1$, $n = m + 1$ and $m' = n/2$) with proposition 3.7.

4.3 An algebraic family

Below we give a previously unpublished construction of $\Omega(p^2)$ four-dimensional $\mathbb{F}_p^*$-matching vectors modulo a prime p. Later in lemma 4.20 we establish its asymptotic optimality. The technique here is different from the techniques used in sections 4.1 and 4.2. The resulting family however is not bounded, therefore it does not immediately imply locally decodable codes capable of tolerating a constant fraction of errors.

Lemma 4.12. Let p be a prime and n be a positive integer. Let π be an $\mathbb{F}_p$-hyperplane in $\mathbb{F}_{p^n}$ and let G be a multiplicative subgroup of $\mathbb{F}_{p^n}^*$. Suppose $|\pi \cap G| = d$; then there exists a $k = \lfloor |G|/d \rfloor$-sized family of $\mathbb{F}_p^*$-matching vectors in $\mathbb{F}_p^n$.

Proof. Let $\phi : \mathbb{F}_{p^n} \to \mathbb{F}_p$ be a linear map such that $\ker \phi = \pi$. Note that there exist exactly d elements $x \in G$ such that $\phi(x) = 0$. Consider a bilinear map $\Phi : \mathbb{F}_{p^n} \times \mathbb{F}_{p^n} \to \mathbb{F}_p$, such that for all y, z,

$$\Phi(y, z) = \phi(y \cdot z).$$

Note that for every $y \in G$ there exist exactly d elements $z \in G$ such that $\Phi(y, z) = 0$. Fix a basis of $\mathbb{F}_{p^n}$ over F_p and represent the map Φ in the coordinate form $\Phi : \mathbb{F}_p^n \times \mathbb{F}_p^n \to \mathbb{F}_p$. This yields a matrix $M \in \mathbb{F}_p^{n \times n}$ such that for every vector $\mathbf{y} \in G \subseteq \mathbb{F}_p^n$ there exist exactly d vectors $\mathbf{z} \in G$ such that

$$\mathbf{y} \cdot M \cdot \mathbf{z}^t = 0.$$

For every vector $\mathbf{g} \in G$ set $\mathbf{u_g} = \mathbf{g}$ and $\mathbf{v_g} = M \cdot \mathbf{g}^t$. Now for every vector in $\{\mathbf{u_g}\}_{\mathbf{g} \in G}$ there exist exactly d vectors in $\{\mathbf{v_h}\}_{\mathbf{h} \in G}$ such that

$$(\mathbf{u_g}, \mathbf{u_h}) = 0.$$

We apply the greedy procedure to the two families of vectors above to obtain new families $\mathcal{U}, \mathcal{V}$ where for each $\mathbf{u} \in \mathcal{U}$ there exists a unique $\mathbf{v} \in \mathcal{V}$ such that $(\mathbf{u}, \mathbf{v}) = 0$. □

Lemma 4.13. Let p be an odd prime. Then the hyperplane

$$\pi = \{x \in \mathbb{F}_{p^4} \mid x - x^p + x^{p^2} - x^{p^3} = 0\}$$

and the multiplicative group

$$G = \{x \in \mathbb{F}_{p^4}^* \mid x^{p^2+1} = 1\}.$$

share exactly two elements of $\mathbb{F}_{p^4}$. Namely ± 1.

Proof. First, note that π is indeed a hyperplane. This follows from the fact that π is a kernel of a linear map, whose image is of size p, (since for every z in the image of the polynomial $x - x^p + x^{p^2} - x^{p^3}$ we have $z^p + z = 0$.) Now let $x \in \pi \cap G$. Combining $x^{p^2} = 1/x$ with the equation defining π we conclude

$$(x + 1/x) - (x + 1/x)^p = 0. \tag{4.3}$$

Thus $x + 1/x \in \mathbb{F}_p$. Therefore x satisfies a quadratic equation with coefficients in $\mathbb{F}_p$. Thus $x \in \mathbb{F}_{p^2}$, and $x^{p^2} = x$. Recall that earlier we had $x^{p^2} = 1/x$. Thus $x^2 = 1$. □

Combining lemma 4.12 and lemma 4.13 we get

Theorem 4.14. Let p be an odd prime. There exists a $(p^2+1)/2$-sized family of $\mathbb{F}_p^*$-matching vectors in $\mathbb{F}_p^4$.

4.4 Upper bounds for families modulo primes

We now turn to upper bounds on $k(m, n)$, where $k(m, n)$ denotes the size of the largest family of $(\mathbb{Z}_m \setminus \{0\})$-matching vectors in $\mathbb{Z}_m^n$. Note that there is a body of work in combinatorics on the closely related problem of bounding the size of set systems with restricted modular intersections. The problem there is to bound the size of the largest set family $\mathcal{F}$ on $[n]$, where the sets in $\mathcal{F}$ have cardinality 0 modulo some integer m, while their intersections have non-zero cardinality modulo m. The classical result in this area shows that when m is a prime power an upper bound of $n^{O(m)}$ holds [7]. No such bound applies when m is composite [53]. The best bound for general m is $|\mathcal{F}| \leq 2^{n/2}$ [85].

We start by bounding $k(m, n)$ in the case when $m = p$ is prime and present two bounds. The first bound is based on the linear algebra method [7] and is tight when p is a constant.

Theorem 4.15. For any positive integer n and any prime p, we have

$$k(p, n) \leq 1 + \binom{n + p - 2}{p - 1}.$$

Proof. Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of S-matching vectors of $\mathbb{F}_p^n$, for some $S \subseteq \mathbb{F}_p^*$. For each $i \in [k]$, we consider the polynomial

$$P_i(z_1, \ldots, z_n) = 1 - \left(\sum_{l=1}^{n} \mathbf{v}_i(l) \cdot z_l \right)^{p-1}$$

in the ring $\mathbb{F}_p[z_1, \ldots, z_n]$. It is easy to see that $P_i(\mathbf{u}_i) = 1$, whereas $P_i(\mathbf{u}_j) = 0$ for all $j \neq i$. This implies that the k polynomials $\{P_i\}_{i=1}^k$ are linearly independent. But these polynomials all lie in an $\mathbb{F}_p$ vector-space of dimension $1 + \binom{n+p-2}{p-1}$, since they are spanned by the monomial 1 and all monomials of degree exactly $p-1$ in $z_1, \ldots, z_n$. $\square$

Note that equation (4.2) shows that for constant p and growing n, the above bound is asymptotically tight.

Our second bound comes from translating the problem of constructing matching vectors into a problem about points and hyperplanes in projective space. The $n-1$ dimensional projective geometry $\mathrm{PG}(\mathbb{F}_\mathrm{p}, \mathrm{n}-1)$ over $\mathbb{F}_p$ consist of all points in $\mathbb{F}_p^n \setminus \{0^n\}$ under the equivalence relation $\lambda\mathbf{v} \equiv \mathbf{v}$ for $\lambda \in \mathbb{F}_p^*$. Projective hyperplanes are specified by vectors $\mathbf{u} \in \mathbb{F}_p^n \setminus \{0^n\}$ under the equivalence relation $\lambda\mathbf{u} \equiv \mathbf{u}$ for $\lambda \in \mathbb{F}_p^*$; such a hyperplane contains all points $\mathbf{v}$ where $(\mathbf{u}, \mathbf{v}) = 0$.

We define a bipartite graph $H(U, V)$ where the vertices on the left correspond to all hyperplanes in $\mathrm{PG}(\mathbb{F}_\mathrm{p}, \mathrm{n}-1)$, vertices on the right correspond to all points in $\mathrm{PG}(\mathbb{F}_\mathrm{p}, \mathrm{n}-1)$ and $\mathbf{u}$ and $\mathbf{v}$ are adjacent if $(\mathbf{u}, \mathbf{v}) = 0$. For $X \subseteq U$ and $Y \subseteq V$, we define $N(X)$ and $N(Y)$ to be their neighborhoods. We use $N(\mathbf{u})$ for the neighborhood of $\mathbf{u}$.

Definition 4.16. Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG}(\mathbb{F}_\mathrm{p}, \mathrm{n}-1)$. We say that a set $X \subseteq U$ satisfies the *unique neighbor property* if for every $\mathbf{u} \in X$, there exists $\mathbf{v} \in N(\mathbf{u})$ such that $\mathbf{v}$ is not adjacent to $\mathbf{u}'$ for any $\mathbf{u}' \in X \setminus \{\mathbf{u}\}$.

Lemma 4.17. Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG}(\mathbb{F}_\mathrm{p}, \mathrm{n}-1)$. There exists a set $X \subseteq U$,

$|X| = k$ satisfying the unique neighbor property if and only if there exists a k-sized family of $\mathbb{Z}_p^*$-matching vectors in $\mathbb{Z}_p^n$.

Proof. Assume that $X = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ satisfies the unique neighbor property. Let $Y = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be such that $\mathbf{v}_i$ is a unique neighbor of $\mathbf{u}_i$. This implies that $(\mathbf{u}_i, \mathbf{v}_i) = 0$ and $(\mathbf{u}_j, \mathbf{v}_i) \neq 0$ for $i \neq j$. Thus X, Y gives a $\mathbb{Z}_p^*$-matching vector family in $\mathbb{Z}_p^n$.

For the converse, let us start with a k-sized matching vector family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_p^n$. In case $k = 1$ the lemma holds trivially. We claim that if $k \geq 2$; then $\mathbf{u} \in \mathcal{U}$ implies that $\lambda\mathbf{u} \notin \mathcal{U}$ for any $\lambda \in \mathbb{F}_p^* \setminus \{1\}$. This is true since $(\mathbf{u}, \mathbf{v}) = 0$ implies $(\lambda\mathbf{u}, \mathbf{v}) = 0$, which would violate the definition of a matching vector family. Thus we can associate each $\mathbf{u} \in \mathcal{U}$ with a distinct hyperplane in $\mathrm{PG(\mathbb{F}_p, n-1)}$. Similarly, we can associate every $\mathbf{v} \in \mathcal{V}$ with a distinct point in $\mathrm{PG(\mathbb{F}_p, n-1)}$. It is easy to see that $\mathbf{v}_i$ is a unique neighbor of $\mathbf{u}_i$, hence the set $\mathcal{U}$ satisfies the unique neighbor property. □

Corollary 4.18. Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG(\mathbb{F}_p, n-1)}$. The size of the largest set $X \subseteq U$ that satisfies the unique neighbor property is exactly $k(p, n)$.

The expansion of the graph $H(U, V)$ was analyzed by Alon using spectral methods [1, theorem 2.3]. We use the rapid expansion of this graph to bound the size of the largest matching vector family.

Lemma 4.19. Let $n \geq 2$ be an integer and p be a prime. Let U (V) be the set of hyperplanes (points) in $\mathrm{PG(\mathbb{F}_p, n-1)}$. Let $u = \frac{p^n-1}{p-1} = |U| = |V|$. For any nonempty set $X \subseteq U$ with $|X| = x$,

$$|N(X)| \geq u - u^{\frac{n}{n-1}}/x. \tag{4.4}$$

Lemma 4.20. Let n be a positive integer and p be a prime; then

$$k(p, n) \leq 4p^{n/2} + 2. \tag{4.5}$$

Proof. If $n = 1$, inequality (4.5) holds trivially. We assume $n \geq 2$. Let $\mathcal{U} \subseteq U$, $\mathcal{V} \subseteq V$ be a matching family of size $k(p, n)$. Pick $X \subseteq \mathcal{U}$ of size $x > 0$. By formula (4.4),

$$|N(X)| \geq u - u^{\frac{n}{n-1}}/x.$$

Since every point in $\mathcal{U} \setminus X$ must contain a unique neighbor from the set $V \setminus N(X)$, we have

$$|\mathcal{U} \setminus X| \leq |V \setminus N(X)| \leq \frac{u^{\frac{n}{n-1}}}{x} \;\Rightarrow\; |\mathcal{U}| \leq \frac{u^{\frac{n}{n-1}}}{x} + x. \tag{4.6}$$

Note that the inequality in the right hand side of (4.6) holds for all positive integers x. Picking $x = \left\lceil u^{\frac{n}{2(n-1)}} \right\rceil$ gives

$$\begin{aligned} |\mathcal{U}| &\leq 2 \left\lceil u^{\frac{n}{2(n-1)}} \right\rceil \leq 2 \left(\frac{p^n}{p-1} \right)^{\frac{n}{2(n-1)}} + 2 = \\ &= 2 \left(\frac{p}{p-1} \right)^{\frac{n}{2(n-1)}} p^{n/2} + 2 \leq 4p^{n/2} + 2, \end{aligned}$$

where the last inequality is a simple calculation. □

Equation (4.1) shows that $k(p, n) = \Omega\left(p^{(n-3)/2}\right)$, so the above upper bound is nearly tight when n is a constant and p grows to infinity. Note that for this setting of parameters, the linear-algebra bound gives $k(p, n) \leq O(p^{n-1})$, so the bound above gives a significant improvement.

4.5 Upper bounds for families modulo prime powers

Bounds for matching families modulo prime powers are obtained via a reduction to the prime case.

Lemma 4.21. Let n be a positive integer, p be a prime and $e \geq 2$. We have

$$k(p^e, n) \leq p^{(e-1)n} k(p, n+1).$$

Proof. Assume for contradiction that we have a matching family $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}, \mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of size $k > p^{(e-1)n} k(p, n+1)$ in $\mathbb{Z}_{p^e}^n$. For every $i \in [k]$, write $\mathbf{u}_i = \mathbf{u}'_i + p^{e-1}\mathbf{u}''_i$ where $\mathbf{u}'_i \in \mathbb{Z}^n_{p^{e-1}}$ and $\mathbf{u}''_i \in \mathbb{Z}^n_p$. By the pigeonhole principle, there are $k' > k(p, n+1)$ values of i which give the same vector $\mathbf{u}'_i \in \mathbb{Z}^n_{p^{e-1}}$, assume for convenience that the corresponding vectors in $\mathcal{U}$ are $\mathbf{u}_1, \ldots, \mathbf{u}_{k'}$ with matching vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k'}$. We will use these vectors to construct a matching vector family of size $k' > k(p, n+1)$ in $\mathbb{Z}^{n+1}_p$, which gives a contradiction.

For each $i \in [k']$, we extend $\mathbf{u}''_i$ to a vector $\bar{\mathbf{u}}_i$ by appending 1 in the last coordinate. For every $i \in [k']$, write $\mathbf{v}_i = \mathbf{v}'_i + p\mathbf{v}''_i$ where $\mathbf{v}'_i \in \mathbb{Z}^n_p$ and $\mathbf{v}''_i \in \mathbb{Z}^n_{p^{e-1}}$. We extend $\mathbf{v}'_i$ to a vector $\bar{\mathbf{v}}_i$ by appending $(\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \in \mathbb{Z}_p$ in the last coordinate (we will show that this ratio is in fact integral).

We claim that for all $i \in [k']$, $(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = 0 \bmod p$. To see this, observe that

$$(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = (\mathbf{u}''_i, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1}. \tag{4.7}$$

But we have

$$\begin{aligned}(\mathbf{u}_i, \mathbf{v}_i) = (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_i, \mathbf{v}_i) \equiv \\ \equiv (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_i, \mathbf{v}'_i) = 0 \bmod p^e.\end{aligned}$$

From this we conclude that $(\mathbf{u}'_i, \mathbf{v}_i) \equiv 0 \bmod p^{e-1}$, and that $(\mathbf{u}''_i, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} = 0 \bmod p$. From equation (4.8), we conclude that $(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = 0 \bmod p$. Next we claim that $(\bar{\mathbf{u}}_j, \bar{\mathbf{v}}_i) \neq 0 \bmod p$ for $i \neq j \in [k']$. We have

$$(\bar{\mathbf{u}}_j, \bar{\mathbf{v}}_i) = (\mathbf{u}''_j, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \tag{4.8}$$

But, since $\mathbf{u}'_i = \mathbf{u}'_j$, we also have

$$\begin{aligned}(\mathbf{u}_j, \mathbf{v}_i) = (\mathbf{u}'_j, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_j, \mathbf{v}_i) \equiv \\ \equiv (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_j, \mathbf{v}'_i) \not\equiv 0 \bmod p^e,\end{aligned}$$

which implies that $(\mathbf{u}''_j, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \not\equiv 0 \bmod p$. This shows that the vectors $\{\bar{\mathbf{u}}_j\}_{j=1}^{k'}$, $\{\bar{\mathbf{v}}_i\}_{i=1}^{k'}$ give a matching vector family of size $k' > k(p, n+1)$, which is a contradiction. □

4.6 Upper bounds for families modulo composites

Bounds for matching families modulo composites are obtained via reductions to the prime power case.

Lemma 4.22. Let m, n, and q be arbitrary positive integers such that $q|m$ and $(q, m/q) = 1$; then

$$k(m,n) \leq (m/q)^n \, k(q,n).$$

Proof. Let us write $m/q = r$. Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of S-matching vectors of $\mathbb{Z}_m^n$, for some $S \subseteq \mathbb{Z}_m \setminus \{0\}$. For each vector $\mathbf{u} \in \mathbb{Z}_m^n$ we can define the vectors $\mathbf{u}' \equiv \mathbf{u} \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{u}'' \equiv \mathbf{u} \bmod r \in \mathbb{Z}_r^n$. From the definition of a matching vector family, we have that

- For all $i \in [k]$, $(\mathbf{u}'_i, \mathbf{v}'_i) = 0$ and $(\mathbf{u}''_i, \mathbf{v}''_i) = 0$;
- For all $i, j \in [k]$ such that $i \neq j$, $(\mathbf{u}'_j, \mathbf{v}'_i) \neq 0$ or $(\mathbf{u}''_j, \mathbf{v}''_i) \neq 0$.

Assume $k > (m/q)^n \, k(q,n)$. By the pigeonhole principle, there exists a vector $\mathbf{u} \in \mathbb{Z}_r^n$ such that $\mathbf{u}''_j = \mathbf{u}$ holds for $k' > k(q,n)$ values of $j \in [k]$. Let us assume that these values are $1, \ldots, k'$. Note that for any $i, j \in [k']$ we have $(\mathbf{u}''_j, \mathbf{v}''_i) = (\mathbf{u}''_i, \mathbf{v}''_i) = 0$. Hence, by the definition of a matching family, we must have

- For all $i \in [k']$, $(\mathbf{u}'_i, \mathbf{v}'_i) = 0$;
- For all $i, j \in [k']$ such that $i \neq j$, $(\mathbf{u}'_j, \mathbf{v}'_i) \neq 0$.

Thus vectors $\{\mathbf{u}'_1, \ldots, \mathbf{u}'_{k'}\}$ and $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_{k'}\}$ form a matching family mod q of size larger than $k(q,n)$ which gives a contradiction. □

Theorem 4.23. Let m and n be arbitrary positive integers. Suppose p is a prime divisor of m; then

$$k(m,n) \leq 5 \frac{m^n}{p^{(n-1)/2}}.$$

Proof. Let p^e be the largest power of p which divides m. By lemmas 4.22, 4.21 and 4.20, we get

$$k(m,n) \leq \left(\frac{m}{p^e}\right)^n p^{(e-1)n} \left(4p^{(n+1)/2} + 2\right) \leq 5 \frac{m^n}{p^{(n-1)/2}}$$

This concludes the proof. □

The above bound is weak when n and p are constants, for instance it is meaningless for $n = 1$. We give another bound below which handles the case of small m. We start with the case when $n = 1$.

Lemma 4.24. Let $m \geq 2$ be an arbitrary positive integer; then

$$k(m, 1) \leq m^{O(1/\log\log m)} = m^{o_m(1)}.$$

Proof. Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}, \mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of $\mathbb{Z}_m \setminus \{0\}$-matching vectors in $\mathbb{Z}_m^1$. We treat every vector $\mathbf{u} \in \mathcal{U}$ as an integer and observe that for any $i \neq j \in [k]$, $\gcd(\mathbf{u}_i, m) \neq \gcd(\mathbf{u}_j, m)$. (Otherwise $(\mathbf{u}_i, \mathbf{v}_i) = 0$ would yield $(\mathbf{u}_j, \mathbf{v}_i) = 0$.) An application of a standard upper bound on the number of distinct divisors of an integer [56] concludes the proof. □

We now proceed to the case of general n.

Theorem 4.25. Let m and n be arbitrary positive integers; then

$$k(m, n) \leq m^{n-1+o_m(1)}.$$

Proof. Given a vector $\mathbf{u} \in \mathbb{Z}_m^n$, we define the $\mathbb{Z}_m$-orbit of $\mathbf{u}$ to be the set of all vectors that can be written as $\lambda\mathbf{u}$ for $\lambda \in \mathbb{Z}_m$. Unlike over $\mathbb{Z}_p$, these orbits are no longer disjoint. We claim that all of $\mathbb{Z}_m^n$ can be covered by no more $\frac{m^n}{\phi(m)}$ orbits, and that each such orbit can contribute at most $k(m, 1)$ vectors to $\mathcal{U}$.

Let $U \subseteq \mathbb{Z}_m^n$ denote the set of all vectors $\mathbf{u}$ such that the GCD of all coordinates of $\mathbf{u}$ is 1. Any vector $\mathbf{u}' \in \mathbb{Z}_m^n$ can be written it as $\lambda\mathbf{u}$ for $\mathbf{u} \in U$ and $\lambda \in \mathbb{Z}_m$. Thus the orbits of vectors in U cover all of $\mathbb{Z}_m^n$. For $\mathbf{u}, \mathbf{u}' \in U$, we say that $\mathbf{u}' \equiv \mathbf{u}''$ if $\mathbf{u}''$ lies in the $\mathbb{Z}_m$ orbit of $\mathbf{u}'$. It is easy to see that this is indeed an equivalence relation on U, which divides U into equivalence classes of size $\phi(m)$. Thus if we pick $U' \subseteq U$ which contains a single representative of each equivalence class, then the orbits of U' contain all of $\mathbb{Z}_m^n$. Thus we have $|U'| = \frac{|U|}{\phi(m)} \leq \frac{m^n}{\phi(m)}$.

Now consider the orbit of any vector $\mathbf{u}$. Assume that it contributes the vector $\mathbf{u}_1 = \lambda_1\mathbf{u}, \ldots, \lambda_t\mathbf{u}$ to $\mathcal{U}$ where $\lambda_i \in \mathbb{Z}_m$. Assume that the matching vectors in $\mathcal{V}$ are $\mathbf{v}_1, \ldots, \mathbf{v}_t$. Then it is easy to see that $\mathcal{U}' = \{\lambda_1, \ldots, \lambda_t\}$ and $\mathcal{V}' = \{(\mathbf{u}, \mathbf{v}_1), \ldots, (\mathbf{u}, \mathbf{v}_t)\}$ are a matching vector family in one dimension, so that $t \leq k(m, 1)$. Thus we conclude that

$$k(m,n) \leq \frac{m^n}{\phi(m)} k(m,1) \leq m^{n-1+o_m(1)}.$$

using a standard lower bound on $\phi(m)$ [56] and lemma 4.24. □

5

Lower bounds

In this chapter we review existing lower bounds for the codeword length of general locally decodable codes. The proofs of these bounds fit the following high level strategy. Firstly, one converts a locally decodable code into a normal form where the decoder is restricted to operate by outputting a modulo 2 sum of some r codeword coordinates coming from a family of disjoint r-tuples. Secondly, one argues that any code presented in a normal form requires a large codeword length.

In section 5.1 we deal with the conversion to the normal form. In section 5.2 we establish polynomial lower bounds for the codeword length of r-query codes for general r. The bound rapidly deteriorates as r increases. In section 5.3 we deal with the narrow case of 2-query codes and establish tight exponential lower bounds. Throughout the chapter we restrict our attention to binary codes of constant query complexity.

5.1 Preliminaries

General locally decodable codes can be quite complex. Decoders may invoke complicated adaptive procedures to decide which codeword bits to query. They may also perform arbitrary computation to come up with the output. In order to prove lower bounds for locally decodable codes it is convenient to first turn them into the following normal form.

Definition 5.1. A binary code $C : \mathbb{F}_2^k \to \mathbb{F}_2^N$ is said to be (r, η, β)-normally decodable if for each $i \in [k]$ there a collection M_i of $\eta \cdot N$ disjoint tuples of exactly r indices from $[N]$ such that for every $t \in M_i$ the following holds

$$\Pr_{\mathbf{x}\in\mathbb{F}_2^{\mathrm{n}}}\left[\mathbf{x}_{\mathrm{i}} = \sum_{\mathrm{j}\in\mathrm{t}} \mathrm{C}(\mathbf{x})_{\mathrm{j}}\right] \geq \frac{1}{2} + \beta, \tag{5.1}$$

where the probability is taken uniformly over $\mathbf{x}$.

Hence to decode $\mathbf{x}_i$ from $C(\mathbf{x})$, the decoder can just add up the indices in a randomly chosen tuple t from M_i. Note that normally decodable codes are somewhat weaker objects than usual locally decodable codes. Specifically, normally decodable codes only provide an “average-case” guarantee of correct decoding. Our main goal in this section is to prove the following lemma from [64].

Lemma 5.2. Suppose there exists a (r, δ, ϵ)-locally decodable code encoding k-bit messages to N-bit codewords where $\epsilon < 1/2$; then there exists a (r, η, β)-normally decodable code encoding k-bit messages to $O(N)$-bit codewords where

$$\eta \geq \frac{(1/2 - \epsilon)\delta}{3 \cdot r^2 2^{r-1}} \quad \text{and} \quad \beta \geq \frac{1/2 - \epsilon}{2^{2r}}.$$

Proof. Our proof proceeds in four steps. On the first step we turn a potentially adaptive decoder of the code C into a non-adaptive one, i.e., a one that makes all queries to the codeword simultaneously. On the second step we turn the locally decodable code into a smooth one, i.e., a one where no codeword coordinate in queried too often. On the third step, we ensure that r-tuples of coordinates that may be read by the decoder interested in the i-th message bit are all disjoint. Finally on the fourth step, we ensure that the decoder always returns a modulo 2 sum of the accessed codeword coordinates. On each step we incur a certain loss in code parameters. Let $\alpha = 1/2 - \epsilon$ be the advantage of the local decoder of the code C over random guessing.

Step 1: Let $\mathcal{A}$ be the potentially adaptive local decoder for C. We now construct a non-adaptive local decoder $\mathcal{A}'$ for the same code C at a price of reducing the value of α to $\alpha/2^{r-1}$. The decoder $\mathcal{A}'$ guesses the values of the first $r-1$ coordinates that may be accessed by $\mathcal{A}$, and submits the set of queries based upon this guess. If $\mathcal{A}'$ guesses correctly the decoding procedure works with probability $1/2 + \alpha$; otherwise, $\mathcal{A}'$ returns a random bit which is correct with probability $1/2$.

Step 2: We now adjust the nonadaptive r-query decoding procedure $\mathcal{A}$ that we got from the previous step to obtain a new decoding procedure $\mathcal{A}'$ such that for all $\mathbf{x} \in \mathbb{F}_2^k$ and $i \in [k]$, we have

$$\mathrm{Pr}\left[\mathcal{A}'(\mathrm{C}(\mathbf{x}), \mathrm{i}) = \mathbf{x}_\mathrm{i}\right] \geq 1/2 + \alpha/2^{\mathrm{r}-1}, \tag{5.2}$$

and for every $i \in [k]$ and $j \in [N]$,

$$\mathrm{Pr}\left[\mathcal{A}'(\cdot, \mathrm{i}) \text{ reads index } j\right] \leq \mathrm{r}/\delta\mathrm{N}. \tag{5.3}$$

For every $i \in [k]$, let $S_i \subseteq [N]$ denote the set of codeword coordinates that are accessed by $\mathcal{A}$ on an input i with probability above $r/\delta N$. Since $\mathcal{A}$ reads at most r indices in every invocation, for every $i \in [k]$, we have $|S_i| \leq \delta \cdot N$. We define the new decoder $\mathcal{A}'$ as follows: $\mathcal{A}'(\cdot, i)$ runs $\mathcal{A}(\cdot, i)$ in a black-box manner by reading indices from the codeword, and returning their values to $\mathcal{A}$. The only exception is that if $\mathcal{A}$ requests an index in S_i, $\mathcal{A}'$ does not read that index, but instead simply returns 0 to $\mathcal{A}$. Thus the output of $\mathcal{A}'$ on $C(\mathbf{x})$ is the same as the output of $\mathcal{A}$ on a certain string $\mathbf{y}$ such that $d(C(\mathbf{x}), \mathbf{y}) \leq \delta N$. It remains to note that given access to any such string $\mathcal{A}$ outputs $\mathbf{x}_i$ with probability at least $1/2 + \alpha/2^{r-1}$.

Step 3: We now modify the decoding procedure $\mathcal{A}$ that we got from the previous step to ensure that for every i, the tuples of coordinates that may be read by the decoder interested in the i-the message bit are all disjoint. Fix $i \in [k]$. Let $S \subseteq [N]$, $|S| \leq r$ be arbitrary. We say that S is γ-good if

$$\mathrm{Pr}_\mathbf{x}\left[\mathcal{A}(\mathrm{C}(\mathbf{x}, \mathrm{i})) = \mathbf{x}_\mathrm{i} \mid \mathcal{A} \text{ reads coordinates in } S\right] \geq 1/2 + \gamma. \tag{5.4}$$

Consider a hypergraph H that contains N vertices labeled by elements of $[N]$. The hyperedges of H, denoted E are defined by

$$E = \{e \subseteq [N] \mid e \text{ is } \alpha/2^r\text{-good}\}.$$

We now argue that the probability that $\mathcal{A}(\cdot, i)$ reads an edge from E is at least $\alpha/2^{r-1}$. To see this note that by formula (5.2)

$$\begin{array}{rr} 1/2 + \alpha/2^{r-1} & \leq \\ \mathrm{Pr}_{\mathbf{x}}\left[\mathcal{A}(\mathrm{C}(\mathbf{x}), \mathrm{i}) = \mathbf{x}_{\mathrm{i}} \mid \mathcal{A}(\cdot, i) \text{ reads } E\right] \cdot \Pr\left[\mathcal{A}(\cdot, \mathrm{i}) \text{ reads } E\right] & + \\ \mathrm{Pr}_{\mathbf{x}}\left[\mathcal{A}(\mathrm{C}(\mathbf{x}), \mathrm{i}) = \mathbf{x}_{\mathrm{i}} \mid \mathcal{A}(\cdot, i) \text{ reads } E^c\right] \cdot \Pr\left[\mathcal{A}(\cdot, \mathrm{i}) \text{ reads } E^c\right] & \leq \\ \mathrm{Pr}_{\mathbf{x}}\left[\mathcal{A}(\cdot, i) \text{ reads } E\right] + (1/2 + \alpha/2^{\mathrm{r}}) \cdot (1 - \Pr\left[\mathcal{A}(\cdot, \mathrm{i}) \text{ reads } E\right]). & \end{array}$$

For each hyperedge $e \in E$ let p_e denote the probability that $\mathcal{A}(\cdot, i)$ reads e. The argument above implies that

$$\sum_{e \in E} p_e \geq \alpha/2^{r-1}.$$

Furthermore, for every $j \in [N]$ formula (5.3) yields

$$\sum_{e \in E \ \mid \ j \in e} p_e \leq r/\delta N.$$

Let V be a vertex cover for the hypergraph H. Since for every $e \in E$ we have $e \cap V \neq 0$, it follows that

$$\sum_{e \in E \ \mid \ e \cap V \neq 0} p_e \geq \alpha/2^{r-1}.$$

Therefore

$$\alpha/2^{r-1} \leq \sum_{e \in E \ \mid \ e \cap V \neq 0} p_e \leq \sum_{j \in V} \sum_{e \in E \ \mid \ j \in e} p_e \leq |V| r/\delta N,$$

which implies that the minimum vertex cover for H has size at least $m = \alpha\delta N/r2^{r-1}$. Recall that every hyperedge in H has cardinality at most r. An application of a standard graph theory result implies that H contains a matching M, i.e., a collection of disjoint edges of size at least $|M| \geq m/r = \alpha\delta N/r^2 2^{r-1}$.

We define the new decoder $\mathcal{A}'$ as follows: on input i, $\mathcal{A}'$ picks one of the edges in the matching M uniformly at random, reads the corresponding codeword coordinates and runs $\mathcal{A}(\cdot, i)$ in a black box manner.

Step 4: We first adjust the code C (by making it sometimes read some extra coordinates) and the decoding procedure $\mathcal{A}$ (by fixing some randomness) to ensure that for all $i \in [k]$, $\mathcal{A}(\cdot, i)$ operates by randomly choosing a tuple t of *exactly* r codeword coordinates coming from a matching M_i, and then applying a *deterministic* function to $f_{i,t}(C(\mathbf{x})|_t)$ to obtain the output. For all $i \in [k]$ and $t \in M_i$ we have

$$\mathrm{Pr}_{\mathbf{x}\in\mathbb{F}_2^{\mathrm{n}}}\left[\mathbf{x}_{\mathrm{i}} = \mathrm{f}_{\mathrm{i,t}}\left(\mathrm{C}(\mathbf{x})|_{\mathrm{t}}\right)\right] \geq 1/2 + \alpha/2^{\mathrm{r}}. \tag{5.5}$$

In what follows we modify the decoder $\mathcal{A}$ to ensure that for all indices i and tuples $t \in M_i$, the function $f_{i,t}$ is simply a modulo 2 sum.

Fix some $i \in [k]$ and $t \in M_i$. Consider a function $f = f_{i,t}$. Let $(c_1, \ldots, c_r)$ be the restriction of a codeword $C(\mathbf{x})$ to coordinates in t. Switching from the $\{0,1\}$ notation to the $\{1,-1\}$ notation allows yields

$$\mathbb{E}_{\mathbf{x}}\left[f(c_1, \ldots, c_r) \cdot \mathbf{x}_i\right] \geq \alpha/2^{r-1}.$$

Representing f in the Fourier basis we get

$$\frac{1}{2^r}\mathbb{E}_{\mathbf{x}}\left[\sum_{\chi} \hat{f}(\chi) \cdot \chi(c_1, \ldots, c_t) \cdot \mathbf{x}_i\right] \geq \alpha/2^{r-1}.$$

Equivalently,

$$\sum_{\chi} \hat{f}(\chi)/2^r \cdot \mathbb{E}_{\mathbf{x}}\left[\chi(c_1, \ldots, c_t) \cdot \mathbf{x}_i\right] \geq \alpha/2^{r-1}.$$

Observe that for all $\chi \in \hat{\mathbb{F}}_2^k$ we have $|\hat{f}(\chi)/2^r| \leq 1$. Therefore there exists a character $\chi \in \hat{\mathbb{F}}_2^k$ such that

$$\mathbb{E}_{\mathbf{x}}\left[\chi(c_1, \ldots, c_t) \cdot \mathbf{x}_i\right] \geq \alpha/2^{2r-1}.$$

Returning to the $\{0,1\}$ notation, for some set $S \subseteq [r]$ we must have either

$$\mathrm{Pr}_{\mathbf{x}\in\mathbb{F}_2^{\mathrm{k}}}\left[\mathbf{x}_{\mathrm{i}} = \sum_{\mathrm{j}\in\mathrm{S}} \mathrm{C}(\mathbf{x})_{\mathrm{j}}\right] \geq \frac{1}{2} + \alpha/2^{\mathrm{r}},$$

or

$$\mathrm{Pr}_{\mathbf{x}\in\mathbb{F}_2^{\mathrm{k}}}\left[\bar{\mathbf{x}}_{\mathrm{i}} = \sum_{\mathrm{j}\in\mathrm{S}} \mathrm{C}(\mathbf{x})_{\mathrm{j}}\right] \geq \frac{1}{2} + \alpha/2^{\mathrm{r}},$$

Replacing every coordinate c of $C(\mathbf{x})$ with a triple $\{0, c, \bar{c}\}$, we bring the decoder to the normal form. For each $i \in [k]$ the decoder operates by picking one of r-tuples of coordinates from a matching M_i at random, and outputting the modulo 2 sum. It is not hard to verify that our construction yields matchings of size at least $(1/2 - \epsilon)\delta N/3 \cdot r^2 2^{r-1}$. The advantage over random guessing is at least $\alpha/2^{2r}$. □

5.2 Polynomial lower bound for r-query codes

In this section we prove an $\Omega\left(k^{r/(r-1)}\right)$ lower bound for the codeword length of an arbitrary r-query locally decodable code due to Katz and Trevisan [64]. Somewhat stronger lower bounds of $\tilde{\Omega}\left(k^{1+1/(\lceil r/2 \rceil - 1)}\right)$ have been obtained in [96, 97]. The main idea of the proof is that of a random restriction. We show that if a locally decodable code C is short, then a restriction of C to a randomly chosen small subset of coordinates carries too much information about the message.

Let $\mathcal{H}(\cdot)$ denote the standard entropy function. We need the following information theory lemma.

Lemma 5.3. Let $C : \mathbb{F}_2^k \to D$ be an arbitrary function. Assume there exists a randomized algorithm $\mathcal{A}$ such that for all $i \in [k]$,

$$\mathrm{Pr}_{\mathbf{x}}\left[\mathcal{A}(\mathrm{C}(\mathbf{x}), \mathrm{i}) = \mathbf{x}_{\mathrm{i}}\right] \geq \frac{1}{2} + \beta,$$

where the probability is taken over the random coins of $\mathcal{A}$ as well as over all strings $\mathbf{x}$; then

$$\log |D| \geq (1 - \mathcal{H}(1/2 + \beta))k.$$

Proof. Let $I(\mathbf{x}; C(\mathbf{x}))$ denote the mutual information between $\mathbf{x}$ and $C(\mathbf{x})$. We have

$$I(\mathbf{x}; C(\mathbf{x})) \leq \mathcal{H}(C(\mathbf{x})) \leq \log |D|.$$

Note that we also have

$$\begin{aligned} I(\mathbf{x}; C(\mathbf{x})) &= \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|C(\mathbf{x})) \\ &\geq \mathcal{H}(\mathbf{x}) - \sum_{i}^{k} \mathcal{H}(\mathbf{x}_i|C(\mathbf{x})) \\ &\geq (1 - \mathcal{H}(1/2 + \beta))k. \end{aligned}$$

Combining the inequalities above completes the proof. □

We are now ready to establish

Theorem 5.4. Suppose there exists an (r, δ, ϵ)-locally decodable code encoding k-bit messages to N-bit codewords; then we necessarily have

$$N \geq \Omega\left(\left(\frac{(1/2-\epsilon)\delta}{r^2}\right)^{1/(r-1)}\left(\left(1 - \mathcal{H}\left(\frac{1}{2} + \frac{1/2-\epsilon}{2^{2r}}\right)\right)\cdot k\right)^{r/(r-1)}\right).$$

provided that k is sufficiently large.

Proof. Assume the contrary. Then for infinitely many k we have codes violating the inequality from the theorem statement. Consider such a code C. Apply lemma 5.2 to turn C into a normal form. This yields an (r, η, β)-normally decodable code, where

$$\eta \geq \frac{(1/2-\epsilon)\delta}{3 \cdot r^2 2^{r-1}} \quad \text{and} \quad \beta \geq \frac{1/2-\epsilon}{2^{2r}}.$$

Let $\{M_i\}, i \in [k]$ be the collection of k matchings used by the decoder of C. Let α be a constant to be fixed later. Pick a set $S \subseteq [N]$ at random, including every element of $[N]$ into S with probability $\alpha k/N$. Let y be the random variable counting the number of matchings $\{M_i\}, i \in [k]$ that have at least one hyperedge completely contained in S. It is not hard to verify that

$$\mathbb{E}[y] \geq \left[1 - \left[1 - \left(\frac{\alpha k}{N}\right)^r\right]^{\eta N}\right]\cdot k \geq \left[1 - \left(\frac{1}{e}\right)^{\eta(\alpha k)^r/N^{r-1}}\right]\cdot k.$$

Since C violates the inequality from the theorem statement we have $N = O_{r,\delta,\epsilon}\left(k^{r/(r-1)}\right)$. Thus the right hand side of the inequality above is at least $\Omega_{r,\delta,\epsilon}(k)$. Note that the random variable y takes non-negative integer values up to k. Therefore there is a positive constant probability that y is larger than $\mathbb{E}[y]/2$. Also note that by the Chernoff bound the probability that $|S| > 2\alpha k$ is exponentially small in k. Thus there exists a set $S \subseteq [N]$ such that $|S| \leq 2\alpha k$ and S contains a hyperedge from at least

$$m = 0.5 \cdot \left[1 - \left(\frac{1}{e}\right)^{\eta(\alpha k)^r/N^{r-1}}\right] \cdot k$$

distinct matchings $\{M_i\}, i \in [k]$. This implies that the restriction of C to coordinates in S allows one to make $(1/2 + \beta)$-accurate predictions about m coordinates of $\mathbf{x}$. Be lemma 5.3 we necessarily have

$$\left[1 - \left(\frac{1}{e}\right)^{\eta(\alpha k)^r/N^{r-1}}\right] \cdot (1 - \mathcal{H}(1/2 + \beta)) \cdot k \leq 4\alpha k$$

Setting $\alpha = (1 - \mathcal{H}(1/2 + \beta))$ and making some basic manipulations we obtain

$$N \geq \Omega\left(k^{r/(r-1)} \cdot \eta^{1/(r-1)} \cdot \alpha^{r/(r-1)}\right).$$

Expressing η and α in terms of δ and ϵ we conclude the proof. □

5.3 Exponential lower bound for 2-query codes

In this section we prove an asymptotically tight $2^{\Omega(k)}$ lower bound for the codeword length of an arbitrary 2-query locally decodable code due to Kerenidis and de Wolf [66]. The proof uses quantum information theory. We argue that short 2-query locally decodable codes yield short quantum random access codes and then apply a theorem of Nayak [76] bounding the length of such codes. Our presentation follows [33, 92].

We start with a brief introduction to quantum information theory needed for the proof. A comprehensive treatment of this area can be found in [77].

5.3.1 Quantum information theory

Let n be a positive integer. For our purposes an n-qubit quantum state is vector $\mathbf{q} \in \mathbb{R}^{2^n}$ such that $\sum_{j\in[2^n]} \mathbf{q}_j^2 = 1$. Let $B = \{\mathbf{b}_j\}, j \in [2^n]$ be an orthonormal basis of $\mathbb{R}^{2^n}$. Measuring a quantum state $\mathbf{q}$ in the basis B yields an output $j \in [2^n]$ with probability $(\mathbf{y}, \mathbf{b}_j)^2$.

A quantum *random access code* is an encoding $\mathbf{x} \to \mathbf{q}_\mathbf{x}$ of k-bit strings $\mathbf{x}$ into n-qubit states $\mathbf{q}_\mathbf{x}$, such that any individual bit $\mathbf{x}_i, i \in [k]$ can be recovered with some probability $p \geq 1/2 + \beta$ from $\mathbf{q}_\mathbf{x}$, where the probability is over a uniform choice of $\mathbf{x}$ and the measurement randomness. The following theorem which is a special case of the Holevo bound [57] is due to Nayak [76].

Theorem 5.5. Any encoding $\mathbf{x} \to \mathbf{q}_\mathbf{x}$ of k-bit strings into n-qubit states with recovery probability at least $1/2 + \beta$, necessarily has

$$n \geq (1 - \mathcal{H}(1/2 + \beta))k.$$

5.3.2 Lower bound

We are now ready to establish

Theorem 5.6. If there exists an $(2, \delta, \epsilon)$-locally decodable code C encoding k-bit messages to N-bit codewords; then

$$N \geq 2^{\Omega\left((1/2-\epsilon)^4\delta^2 k\right)}.$$

Proof. Apply lemma 5.2 to turn the code C into a normal form. This yields an $(2, \eta, \beta)$-normally decodable code, where

$$\eta \geq \Omega((1/2 - \epsilon)\delta) \quad \text{and} \quad \beta \geq \Omega(1/2 - \epsilon).$$

We pad the code with zeros to ensure that the codeword length N is a power of two, $N = 2^n$. For every $\mathbf{x} \in \{0,1\}^k$ consider a n-qubit state $\mathbf{q}_\mathbf{x}$, where for all $j \in [N]$,

$$\mathbf{q}_j = (-1)^{C(\mathbf{x})_j}/\sqrt{N}. \tag{5.6}$$

We claim that the map $\mathbf{x} \to \mathbf{q}_\mathbf{x}$ is a quantum random access code. Let $i \in [k]$ be arbitrary. To recover the bit $\mathbf{x}_i$ from the quantum state $\mathbf{q}_\mathbf{x}$, we make a measurement in a suitable basis. Let $\mathbf{e}_m$ denote the m-th unit vector in $\mathbb{R}^N$, and let $M_i = \left\{(c_1^\ell, c_2^\ell)\right\}_{\ell \in [\eta N]}$ be the matching used by the normal decoder for C. Consider an orthonormal basis $B = \{\mathbf{b}_j\}$ for $\mathbb{R}^N$, where

$$\mathbf{b}_j = \begin{cases} \mathbf{e}_j & \text{if } j \notin \mathrm{supp(M_i)}; \\ \frac{1}{\sqrt{2}}\left(\mathbf{e}_{c_1^\ell} + \mathbf{e}_{c_2^\ell}\right) & \text{if } j = c_1^\ell \text{ for some } \ell; \\ \frac{1}{\sqrt{2}}\left(\mathbf{e}_{c_1^\ell} - \mathbf{e}_{c_2^\ell}\right) & \text{if } j = c_2^\ell \text{ for some } \ell. \end{cases}$$

Observe that

$$(\mathbf{b}_j, \mathbf{q_x})^2 = \begin{cases} 1/N & \text{if } j \notin \mathrm{supp(M_i)}; \\ 2/N & \text{if } j = c_1^\ell \text{ for some } \ell \text{, and } C(\mathbf{x})_{c_1^\ell} \oplus C(\mathbf{x})_{c_2^\ell} = 0; \\ 0 & \text{if } j = c_1^\ell \text{ for some } \ell \text{, and } C(\mathbf{x})_{c_1^\ell} \oplus C(\mathbf{x})_{c_2^\ell} = 1; \\ 2/N & \text{if } j = c_2^\ell \text{ for some } \ell \text{, and } C(\mathbf{x})_{c_1^\ell} \oplus C(\mathbf{x})_{c_2^\ell} = 1; \\ 0 & \text{if } j = c_2^\ell \text{ for some } \ell \text{, and } C(\mathbf{x})_{c_1^\ell} \oplus C(\mathbf{x})_{c_2^\ell} = 0. \end{cases}$$

The decoder for the quantum random access code interested in $\mathbf{x}_i$ measures the state $\mathbf{q_x}$ in the basis B. If the output is $j \notin \mathrm{supp(M_i)}$ it outputs a uniformly random bit; otherwise it outputs the modulo two sum of the two coordinates of $C(\mathbf{x})$ from the matching M_i. Such decoder has an advantage of $\eta\beta$ over random guessing. Thus by theorem 5.5 we must have

$$n \geq (1 - \mathcal{H}(1/2 + \eta\beta)) \cdot k.$$

Expressing η and β in terms of δ and ϵ and using the fact that $1 - \mathcal{H}(1/2 + \tau) = \Theta(\tau^2)$ we conclude the proof. □

The dependence on δ and ϵ in the exponent can be improved to $(1/2 - \epsilon)^2\delta$ [66]. An alternative proof of theorem 5.6 is given [18], using an extension of the Bonami-Beckner hypercontractive inequality. However, that proof still follows the outline of the above quantum-inspired proof, albeit in linear-algebraic language.

6

Applications

In this chapter we discuss three most prominent applications of locally decodable codes, namely, applications to private information retrieval (section 6.1), secure multiparty computation (section 6.2), and lower bounds for arithmetic circuits (section 6.3).

6.1 Private information retrieval

Private Information Retrieval (PIR) schemes are cryptographic protocols designed to safeguard the privacy of database users. They allow clients to retrieve records from public databases while completely hiding the identity of the retrieved records from database owners. The possibility of retrieving database records without revealing their identities to the owner of the database may seem beyond hope. Note, however, that a trivial solution is available: When users want a single record, they can ask for a copy of the whole database. This solution involves enormous communication overhead and is likely to be unacceptable. It turns out that for users who want to keep their privacy fully protected (in the information-theoretic sense), this trivial solution is optimal.

Fortunately, the negative result applies only to databases stored on a single server, rather than those replicated across several servers. In 1995, Chor et al. [27] came up with PIR schemes that enable private retrieval of records from replicated databases, with a nontrivially small amount of communication. In such protocols, users query each server holding the database. The protocol ensures that each individual server (by observing only the query it receives) gets no information about the identity of the items of user interest.

We now make the notion of private information retrieval schemes more concrete. We model database as a k-long q-ary string $\mathbf{x}$ that is replicated between r non-communicating servers. The user holds an index i (which is an integer between 1 and k) and is interested in obtaining the value of the i-th coordinate of $\mathbf{x}$. To achieve this goal, the user tosses some random coins, queries each of the r servers and gets replies from which the desired value can be computed. The query to each server is distributed independently of i therefore each server gets no information about what the user is after. Formally,

Definition 6.1. A r-server private information retrieval protocol is a triplet of non-uniform algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$. We assume that each algorithm is given k as an advice. At the beginning of the protocol, the user $\mathcal{U}$ tosses random coins and obtains a random string rand. Next $\mathcal{U}$ invokes $\mathcal{Q}(i, \mathrm{rand})$ to generate an r-tuple of queries $(\mathrm{que}_1, \ldots, \mathrm{que}_r)$. For $j \in [r]$, $\mathcal{U}$ sends que_j to the server $\mathcal{S}_j$. Each server $\mathcal{S}_j$, $j \in [r]$ responds with an answer $\mathrm{ans}_j = \mathcal{A}(j, \mathbf{x}, \mathrm{que}_j)$. Finally, $\mathcal{U}$ computes its output by applying the reconstruction algorithm $\mathcal{C}(\mathrm{ans_1}, \ldots, \mathrm{ans_r}, \mathrm{i}, \mathrm{rand})$. A protocol as above should satisfy the following requirements:

- **Correctness :** For any k, $\mathbf{x} \in [q]^k$ and $i \in [k]$, $\mathcal{U}$ outputs the correct value of $\mathbf{x}_i$ with probability 1 (where the probability is over the random strings rand).
- **Privacy :** Each server individually learns no information about i. More precisely, we require that for any k and for any $j \in [r]$, the distributions $\mathrm{que}_j(i, \mathrm{rand})$ are identical for all values $i \in [k]$.

The *communication complexity* of a PIR protocol $\mathcal{P}$, is a function of k measuring the total number of bits communicated between the user and the servers, maximized over all choices of $\mathbf{x} \in [q]^k$, $i \in [k]$, and random inputs. The major goal of PIR related research to design r-server private information retrieval schemes with optimal (i.e., the smallest possible) amount of communication for every r.

Following the seminal paper of Chor et al. [27] there has been a large a body of work on private information retrieval [4, 13, 14, 100, 96, 101, 81, 38, 61]. A large number of extensions of the basic PIR model have also been studied. These include extensions to t-private protocols, in which the user is protected against collusions of up to t servers [27, 13, 9]; extensions which protect the servers holding the database in addition to the user, termed symmetric PIR [49, 75]; extensions to computational schemes [67, 22, 69, 47] that only ensure that a server cannot get any information about the user's intensions unless it solves a certain computationally hard problem; and other extensions [15, 16, 24, 31, 48, 79]. In many of those extensions the protocols are obtained by adding some extra layers on top of a basic private information retrieval scheme. Therefore improving parameters of basic private information retrieval schemes yields improvements for many other problems. See [44, 102] for surveys of PIR literature.

The gap between upper and lower bounds for communication complexity of private information retrieval schemes is fairly large. Currently, the most efficient r-server schemes for $r \geq 3$ are obtained through r-query locally decodable codes. Communication complexity of such schemes is roughly logarithmic in the codeword length of corresponding codes. This, for instance, yields 3-server schemes with $\exp\left(\sqrt{\log k \log\log k}\right)$ communication to access a k-bit database [38]. Two server private information retrieval schemes do not rely on LDCs. The most efficient such schemes to date require $O(k^{1/3})$ communication [27]. The best lower bound for the communication complexity of two server PIR is $5 \log k$ due to Wehner and de Wolf [96]. Single server PIR schemes require $\Theta(k)$ communication [27].

In what follows we review existing constructions of private information retrieval schemes in more detail. In section 6.1.1 we discuss the construction of schemes from locally decodable codes and in section 6.1.2

we present a two server scheme based on polynomial interpolation.

6.1.1 From codes to schemes

The following lemma obtains an r-server private information retrieval scheme out of any perfectly smooth r-query locally decodable code, i.e., a code where each decoder's query is distributed perfectly uniformly over the set of codeword coordinates.

Lemma 6.2. Suppose there exists a perfectly smooth q-ary r-query locally decodable code C encoding k-long messages to N-long codewords; then there exists an r-server private information retrieval scheme with $O(r \cdot \log_2(Nq))$ communication to access a q-ary k-long database.

Proof. At the preprocessing stage servers $\mathcal{S}_1, \ldots, \mathcal{S}_r$ encode the k-long database $\mathbf{x}$ with the code C. Next the user $\mathcal{U}$ who is interested in obtaining the value of the i-th coordinate of $\mathbf{x}$, tosses random coins and generates an r-tuple of queries $(\mathrm{que}_1, \ldots, \mathrm{que}_\mathrm{r})$, such that $\mathbf{x}_i$ can be computed from $C(x)_{\mathrm{que}_1}, \ldots, C(x)_{\mathrm{que}_\mathrm{r}}$. For every $j \in [r]$, the user sends the query que_j to the server $\mathcal{S}_j$. Each server $\mathcal{S}_j$ responds with a q-ary value $C(\mathbf{x})_{\mathrm{que}_\mathrm{j}}$. The user combines servers' responses to obtain $\mathbf{x}_i$.

It is straightforward to verify that the protocol above is private since for every $j \in [r]$ the query que_j is uniformly distributed over the set of codeword coordinates. The total communication is given by $r \cdot (\lceil \log_2 N \rceil + \lceil \log_2 q \rceil)$. □

Combining lemma 6.2 with theorem 3.11 we get

Theorem 6.3. For every integer $t \geq 2$, and for all $k \geq 2$, there exists a $3 \cdot 2^{t-2}$-server private information retrieval scheme with

$$\exp_t \left((\log k)^{1/t} (\log \log k)^{1-1/t} \right) -$$

bit communication to access a k-bit database.

6.1.2 Two server private information retrieval

Below we present a two server PIR scheme due to Woodruff et al. [100]. The scheme involves $O\left(k^{1/3}\right)$ communication to access a k-bit database and is arguably the most intuitive among existing two server schemes [27, 13, 100]. The ideas behind the scheme are similar to those behind Reed Muller locally decodable codes (section 2.2).

Let n be an arbitrary positive integer. Set $k = \binom{n}{3}$. In what follows we obtain a 2-server scheme with $O(n)$ bits of communication to access an k-bit database. Pick $\gamma : [k] \to \{0,1\}^n$ to be an arbitrary bijection between the set $[k]$ and the set of n-dimensional $\{0,1\}$-vectors of Hamming weight three. For $i \in [k]$ and $j \in \{1,2,3\}$ let $\gamma(i)_j$ denote the j-th nonzero coordinate of $\gamma(i)$. Given a database $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathbb{F}_2^k$ each server obtains the following polynomial F in the ring $\mathbb{F}_2[z_1, \ldots, z_n]$,

$$F(z_1, \ldots, z_n) = \sum_{i=1}^{k} \mathbf{x}_i \cdot z_{\gamma(i)_1} \cdot z_{\gamma(i)_2} \cdot z_{\gamma(i)_3}.$$

The key properties of the polynomial F are the following:

- F encodes the database: For every $i \in [k]$, $F(\gamma(i)) = \mathbf{x}_i$;
- F has low degree: $\deg f = 3$.

Note that the polynomial F can be naturally treated as a polynomial over $\mathbb{F}_4$. The basic idea behind our private information retrieval scheme is the idea of polynomial interpolation. Suppose the user wants to retrieve the i-th coordinate of the database. Given i, the user obtains the vector $\mathbf{w} = \gamma(i) \in \mathbb{F}_4^n$. Now the user's goal is to recover the value of the polynomial F (held by the servers) at the point $\mathbf{w}$.

Obviously, the user cannot explicitly request the value of F at $\mathbf{w}$ from any of the servers, since such a request would ruin the privacy of the protocol; that is, some server will get to know which database bit the user is after. Instead, the user obtains the value of $F(\mathbf{w})$ indirectly, relying on the rich structure of local dependencies between the evaluations of a cubic polynomial F at multiple points. Specifically, the user randomly selects an affine line $L \in \mathbb{F}_4^n$ containing the point $\mathbf{w}$ and discloses certain points on L to the servers. Each server computes and returns the value of F and the values of partial derivatives of F at the point that it is given. Finally, the user reconstructs the restriction of F to L. In particular the user obtains the desired value $F(\mathbf{w})$. Below is a more formal description.

We use the standard mathematical notation $\frac{\partial F}{\partial z_l}\Big|_{\mathbf{y}}$ to denote the value of the partial derivative [68, 29] of F with respect to a variable z_l at a point $\mathbf{y}$. Let $\lambda_1, \lambda_2 \in \mathbb{F}_4$ be distinct and non-zero. Let $\mathcal{U}$ denote the user and $\mathcal{S}_1, \mathcal{S}_2$ denote the servers. The protocol proceeds as follows,

$\mathcal{U}$	:	Picks $\mathbf{v} \in \mathbb{F}_4^n$ uniformly at random.
$\mathcal{U} \to \mathcal{S}_h$	:	$\mathbf{w} + \lambda_h \mathbf{v}$
$\mathcal{U} \leftarrow \mathcal{S}_h$	:	$F(\mathbf{w} + \lambda_h \mathbf{v}),\ \frac{\partial F}{\partial z_1}\Big|_{\mathbf{w}+\lambda_h\mathbf{v}}, \ldots, \frac{\partial F}{\partial z_m}\Big|_{\mathbf{w}+\lambda_h\mathbf{v}}$

Note that in the protocol above the input of each server $\mathcal{S}_h, h \in \{1, 2\}$ is a uniformly random point in $\mathbb{F}_4^n$. Therefore the protocol is private. It is also easy to verify that both the queries that the user sends to servers and the servers' responses are of length $O(n) = O(k^{1/3})$. (Every query is simply a point in $\mathbb{F}_4^n$. Every response is a list of n values of partial derivatives of F plus the value of F itself.) It remains to show how the user obtains $F(\mathbf{w})$ from the servers' responses.

Consider the line $L = \{\mathbf{w} + \lambda\mathbf{v} \mid \lambda \in \mathbb{F}_4\}$. Let $f(\lambda) = f(\mathbf{w} + \lambda\mathbf{v}) \in \mathbb{F}_4[\lambda]$ be the restriction of F to L. Clearly, $f(\lambda_h) = F(\mathbf{w} + \lambda_h\mathbf{v})$. Thus the user knows the values $\{f(\lambda_h)\}$ for $h \in \{1, 2\}$. This, however, does not suffice to reconstruct the polynomial f, since the degree of f may be up to three. The main observation underlying our protocol is that knowing the values of partial derivatives $\frac{\partial F}{\partial z_1}\Big|_{\mathbf{w}+\lambda_h\mathbf{v}}, \ldots, \frac{\partial F}{\partial z_n}\Big|_{\mathbf{w}+\lambda_h\mathbf{v}}$, the user can reconstruct the value of $f'(\lambda_h)$. The proof is a straightforward application of the chain rule:

$$\frac{\partial f}{\partial \lambda}\bigg|_{\lambda_h} = \frac{\partial F(\mathbf{w} + \lambda\mathbf{v})}{\partial \lambda}\bigg|_{\lambda_h} = \sum_{l=1}^{n} \frac{\partial F}{\partial z_l}\bigg|_{\mathbf{w}+\lambda_h\mathbf{v}} \mathbf{v}_l.$$

Thus the user can reconstruct $\{f(\lambda_h)\}$ and $\{f'(\lambda_h)\}$ for $h \in \{1, 2\}$. Combining this observation with the standard algebraic fact that a cubic univariate polynomial is uniquely determined by its values and derivatives at two points [68], we conclude that the user can reconstruct f and obtain $\mathbf{x}_i = F(\mathbf{w}) = f(0)$.

6.2 Secure multiparty computation

A fundamental result of Ben-Or et al. [19] and Chaum et al. [25] from 1988 asserts that information-theoretic secure multiparty computation is feasible. Specifically, in [19, 25] it is shown that $r \geq 3$ players that are allowed to exchange messages over secure channels, can jointly compute any function of their local inputs while hiding the inputs from each other; i.e., one can always arrange a protocol as to ensure that after performing the joint computation any specific player gets no information about the inputs of other players (apart from the information contained in the value of the function).

In all known protocols for secure multiparty computation the communication complexity of the protocol grows linearly with the circuit size of the function being computed. This results in $2^{\Omega(k)}$ amount of communication for securely computing most of the functions of k-bit inputs. A natural question that was explicitly asked in several papers from the late 1980's and early 1990's [32, 12] is whether *all* functions can be securely computed with only a polynomial (or at least a subexponential) amount of communication in the input length. It was observed by Ishai and Kushilevtiz [58] that this question is closely related to the complexity of private information retrieval schemes.

The construction of private information retrieval schemes given in section 6.1 yields quantitative progress on the question mentioned above (via the reduction of [58]). Specifically, theorem 6.3 implies that a group of 18 or more players can securely compute any function of their k-bit inputs with a total communication of $\exp\left(\sqrt{k \log k}\right)$, for all k.

6.3 Circuit lower bounds

In 1977 Valiant [94] has put forward the following definition.

Definition 6.4. A $k \times N$ matrix G over a field $\mathbb{F}$ is called (r, d)-rigid, if G cannot be written as a sum of two matrices $G = L + S$, where the rank of L is at most r, and S contains at most d non-zero entries in every column.

Valiant [94] showed that if a matrix $G \in \mathbb{F}^{k \times N}$ is $(k/2, k^{\alpha})$-rigid, where $N = O(k)$ and $\alpha > 0$; then the linear transformation from $\mathbb{F}^k$ to $\mathbb{F}^N$ induced by G cannot be computed by a linear arithmetic circuit that simultaneously has size $O(k)$ and depth $O(\log k)$. Valiant's work naturally led to the challenge of constructing *explicit* rigid matrices since such matrices yield explicit linear maps for that we have circuit lower bounds. (It is not hard to verify that a random $k \times k$ matrix is $(k/2, \Omega(k))$-rigid with high probability.) This challenge has triggered a long line of work. For references see [42, 63, 71, 86, 72]. However, after more than three decades of efforts, we are still nowhere close to constructing explicit rigid matrices with parameters needed to get implications in complexity theory. The best explicit family of $(k/2, d)$-rigid matrices of size $k \times N$ has $N(k) = k \cdot \exp(d)$ due to Alon et al. [2].

Recently Dvir [35] suggested an approach to obtaining explicit rigid matrices through establishing lower bounds on the codeword length of linear locally correctable codes. Specifically, Dvir [35] proposed the following

Conjecture 6.5. There exists a field $\mathbb{F}$ and positive constants $\alpha, \beta, \gamma, \epsilon$ such that for sufficiently large k there does not exist a linear $(k^{\alpha}, 1/k^{\beta}, \epsilon)$-locally correctable code of dimension k in $\mathbb{F}^{(1+\gamma)k}$.

Dvir [35] argued that if the conjecture above holds; then any generator matrix G of a certain appropriately chosen Reed Muller code is sufficiently rigid to yield circuit lower bounds. On the high level, the proof proceeds as follows: Suppose G is not rigid. Then, the mapping induced by G can be approximated by a sparse mapping S (a mapping in which each output depends on a small number of inputs) in the sense that there exists a large subspace on which G agrees with S. Next, we observe that this subspace is a locally correctable code, since we can correct each coordinate in a corrupted codeword $\mathbf{y}$ by invoking the local decoder for $\mathbf{y} \cdot G$ and simulating each query to $\mathbf{y} \cdot G$ using a small number of queries to the original string $\mathbf{y}$. Finally, we obtain a contradiction with the conjecture 6.5.

We remark that the setting of code parameters in the conjecture above is somewhat different from the settings that we have addressed in previous chapters. Specifically, the conjecture talks about locally correctable codes of very high rate, tolerating a sub-constant fraction of errors. It also interesting to observe that both Valiant's and Dvir's reductions apply over an arbitrary (not necessarily finite) field.

7

Future directions

In this chapter we list and comment on the most exciting open questions relating to locally decodable codes and private information retrieval schemes.

7.1 3-query locally decodable codes

It is very interesting to determine the optimal length of 3-query codes. The best upper bound to date is $\exp\exp\left(\sqrt{\log k \cdot \log\log k}\right)$ due to Efremenko [38]. The best lower bound is $\tilde{\Omega}(k^2)$ due to Woodruff [97, 99].

A natural approach to improving the upper bound is through the matching vector codes machinery detailed in chapters 3 and 4. This calls for constructing families of S-matching vectors in $\mathbb{Z}_m^n$, for small sets S of size larger than what one gets from the Grolmusz construction. We remark that improving the Grolmusz construction for constant values of m will have significant implications other than improved upper bounds for locally decodable codes, e.g., [53] (if explicit) it will give an explicit family of Ramsey graphs beyond the Frankl-Wilson bound different from [8]. One approach to improving the Grolmusz construction is to improve upper bounds for the degree of polynomial representation of the OR function modulo composites [10, 91].

7.2 r-query locally decodable codes

Again, main questions here relate to the true shape of the trade-off between query complexity and codeword length of locally decodable codes. Currently the improvement that matching vector codes provide over Reed Muller codes rapidly deteriorates with an increase in the number of queries, and vanishes entirely once the query complexity reaches $\log k/(\log\log k)^c$. The next two benchmarks we have here for constructions both come from Reed Muller codes. They are to

- Construct codes with $r = O(\log k)$ and polynomial stretch;
- Construct codes with $r = k^{o(1)}$ and positive rate.

Theorems 3.13 and 3.14 indicate that matching vector are unlikely to help us achieve the second benchmark. At the same time it is quite plausible that MV codes may achieve the first benchmark. This calls for new bounded families of matching vectors in $\mathbb{Z}_m^n$, where m is comparable to (or larger than) n. This regime has almost not been addressed in the past.

It is also interesting to understand the power of matching vector codes for other values of the query complexity. The following conjecture has been made by Dvir et al. [36] in this regard. (Recall that $k(m, n)$ denotes the size of the largest $\mathbb{Z}_m \setminus \{0\}$-matching family in $\mathbb{Z}_m^n$.)

Conjecture 7.1. Let m and n be arbitrary positive integers; then

$$k(m, n) \leq O\left(m^{n/2}\right).$$

By lemma 4.20 the conjecture holds for prime m. If the conjecture holds in general; then any matching vector code must have length $N = \Omega(k^2)$, and thus MV codes are inferior to Reed Muller codes once $r \geq \log^2 k$ by an argument similar to the one in section 3.7.2.

7.3 Locally correctable codes

In chapter 2 we remarked that Reed Muller codes constitute the only class of locally correctable codes known to date. It is interesting to see if there exist shorter codes that are locally correctable. In particular we do not know if matching vector codes (that share many properties of Reed Muller codes) can be made locally correctable.

7.4 Two server private information retrieval

Unlike PIR schemes involving three or more servers, existing two server schemes are not based on locally decodable codes. While a number of different two server schemes are known [27, 4, 13, 60, 14, 100], all of them have the same asymptotic communication complexity of $O\left(k^{1/3}\right)$ as the earliest such schemes proposed in [27]. The best lower bound is $5 \cdot \log k$ due to Wehner and de Wolf [96].

One approach to improving the bounds for the communication complexity of two server private information retrieval has been proposed by Razborov et al. [82] who showed that under some weak technical restriction two server schemes with $O(c)$ communication to access a k-bit database are equivalent to matrices M of size $\exp(c) \times \exp(c)$ with entries from the alphabet $\{x_1, \ldots, x_k, *\}$ such that:

(1) Every variable $x_i, i \in [k]$ appears exactly once in each row and each column of M;
(2) For all 2^k assignments of $\mathbb{F}_2$ values to variables $\{x_i\}_{i \in [k]}$, there is a completion of the matrix, (i.e., assignment of $\mathbb{F}_2$ values to locations containing stars) such that the $\mathbb{F}_2$-rank of the resulting matrix is $O(c)$.

7.5 Private information retrieval with preprocessing

Our review of the state of the art in private information retrieval has concentrated on the most studied aspect of PIR schemes, namely, their communication complexity. Another important aspect of such schemes is the amount of computation that servers need to perform in order to respond to user queries. In fact, it is the overwhelming *computational* *complexity* of PIR schemes, that currently presents the main bottleneck to their practical deployment.

Computational complexity of early private information retrieval schemes has been addressed in [15, 100] where it was shown that preprocessing the database can lead to notable savings. It will be interesting to see further results in this direction as well as address the computational complexity of private information retrieval schemes arising from matching vector codes.

Acknowledgements

We would like to thank Parikshit Gopalan for providing the source file of [52] and helpful discussions regarding this survey.

References

[1] Noga Alon. Eigenvalues, geometric expanders, sorting in rounds, amd Ramsey theory. *Combinatorica*, 6:207–219, 1986.

[2] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *13th International Workshop on Randomization and Computation (RANDOM)*, volume 5687 of Lecture Notes in Computer Science, pages 339–351, 2009.

[3] Noga Alon and Joel Spencer. *The probabilistic method.* 2000.

[4] Andris Ambainis. Upper bound on the communication complexity of private information retrieval. In *32th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1256 of Lecture Notes in Computer Science, pages 401–407, 1997.

[5] Anjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23:365–426, 2003.

[6] Laszlo Babai, Lance Fortnow, Leonid Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *23th ACM Symposium on Theory of Computing (STOC)*, pages 21–31, 1991.

[7] Laszlo Babai and Peter Frankl. *Linear algebra methods in combinatorics.* 1998.

[8] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. In *38th ACM Symposium on Theory of Computing (STOC)*, pages 671–680, 2006.

[9] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. In *International Workshop on Randomization and Computation (RANDOM)*, pages 311–325, 2007.

[10] David A. Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:67–382, 1994.

[11] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *7th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 415 of Lecture Notes in Computer Science, pages 37–48, 1990.

[12] Donald Beaver, Joan Feigenbaum, Joe Kilian, and Phillip Rogaway. Security with low communication overhead. In *International Cryptology Conference (CRYPTO)*, pages 62–76, 1990.

[13] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences*, 71:213–247, 2005.

[14] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-Francios Raymond. Breaking the $O\left(n^{1/(2k-1)}\right)$ barrier for information-theoretic private information retrieval. In *43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 261–270, 2002.

[15] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. In *International Cryptology Conference (CRYPTO)*, volume 1880 of Lecture Notes in Computer Science, pages 56–74, 2000.

[16] Amos Beimel and Yoav Stahl. Robust information theoretic private information retrieval. In *3rd Conference of Security in Communication Networks*, 2002.

[17] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. Local list decoding with a constant number of queries. Electronic Colloquium on Computational Complexity (ECCC) TR10-047, 2010.

[18] Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and ldcs. In *49rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 477–486, 2008.

[19] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1988.

[20] Manuel Blum and Sampath Kannan. Designing programs that check their work. In *21th ACM Symposium on Theory of Computing (STOC)*, pages 86–97, 1989.

[21] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993.

[22] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *International Cryptology Conference (EUROCRYPT)*, volume 1592 of Lecture Notes in Computer Science, pages 402–414, 1999.

[23] Peter J. Cameron. *Combinatorics: topics, techniques, algorithms.* 1994.

[24] Ran Canetti, Yuval Ishai, Ravi Kumar, Michael Reiter, Ronitt Rubinfeld, and Rebecca Wright. Selective private function evaluation with applications to private statistics. In *20th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 293–304, 2001.

[25] David Chaum, Claude Crepeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *20th ACM Symposium on Theory of Computing (STOC)*, pages 11–19, 1988.

[26] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Efficient and error-correcting data structures for membership and polynomial evaluation. In *27th Symposium on Theoretical Aspects of Computer Science (STACS)*, 2010.

[27] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45:965–981, 1998.

[28] Ronald de Wolf. Error-correcting data structures. In *26th Annual Symposium on Theoretical Aspects of Computer Science (STACS 09)*, pages 313–324, 2009.

[29] B.L. Van der Waerden. *Algebra*. 2003.

[30] A. Deshpande, R. Jain, T. Kavitha, S. Lokam, and J. Radhakrishnan. Better lower bounds for locally decodable codes. In *20th IEEE Computational Complexity Conference (CCC)*, pages 184–193, 2002.

[31] Giovanni Di-Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Universal service-providers for private information retrieval. *Journal of Cryptology*, 14:37–74, 2001.

[32] Silvio Micali Donald Beaver and Pillip Rogaway. The round complexity of secure protocols. In *22nd ACM Symposium on Theory of Computing (STOC)*, pages 503–513, 1990.

[33] Andrew Drucker and Ronald de Wolf. Quantum proofs for classical theorems. Arxiv 0910.3376, 2009.

[34] Devdatt P. Dubashi and Alessandro Panconesi. *Concentration of measure for the analysis of algorithms*. 2009.

[35] Zeev Dvir. On matrix rigidity and locally self-correctable codes. 2009.

[36] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. Electronic Colloquium on Computational Complexity (ECCC) TR010-012, 2010.

[37] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2006.

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

[39] Peter Elias. List decoding for noisy channels. Research laboratory for electronics, MIT, 1957.

[40] David Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[41] G. David Forney. *Concatenated codes*. 1966.

[42] Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13:235–239, 1993.

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

[44] William Gasarch. A survey on private information retrieval. *The Bulletin of the EATCS*, 82:72–107, 2004.

[45] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self testing / correcting for polynomials and for approximate functions. In *23th ACM Symposium on Theory of Computing (STOC)*, pages 32–42, 1991.

[46] Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43:169–174, 1992.

[47] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In *32th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 803–815, 2005.

[48] Yael Gertner, Shafi Goldwasser, and Tal Malkin. A random server model for private information retrieval. In *International Workshop on Randomization and Computation (RANDOM)*, volume 1518 of Lecture Notes in Computer Science, pages 200–217, 1998.

[49] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *Journal of Computer and System Sciences*, 60:592–629, 2000.

[50] Oded Goldreich. Short locally testable codes and proofs. Electronic Colloquium on Computational Complexity (ECCC) TR05-014, 2005.

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

[52] Parikshit Gopalan. *Computing with Polynomials over Composites.* PhD thesis, Georgia Institute of Technology, 2006.

[53] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20:71–86, 2000.

[54] Vince Grolmusz. Constructing set-systems with prescribed intersection sizes. *Journal of Algorithms*, 44:321–337, 2002.

[55] Venkatesan Guruswami. *List decoding of error-correcting codes.* PhD thesis, Massachusetts Institite of Technology, 2001.

[56] G. Hardy and E. Wright. *An introduction to the theory of numbers.* 1985.

[57] Alexander Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problems of Information Transmission*, 9:177–183, 1973.

[58] Yuval Ishai and Eyal Kushilevitz. On the hardness of information-theoretic multiparty computation. In *Eurocrypt 2004*, volume 3027 of Lecture Notes in Computer Science, pages 439–455, 2004.

[59] Yuval Ishai and Eyal Kushilevitzy. Improved upper bounds on information theoretic private information retrieval. In *31th ACM Symposium on Theory of Computing (STOC)*, pages 79–88, 1999.

[60] Toshiya Itoh. Efficient private information retrieval. *IEICE Trans. Fund. of Electronics, Commun. and Comp. Sci.*, E82-A:11–20, 1999.

[61] Toshiya Itoh and Yasuhiro Suzuki. New constructions for query-efficient locally decodable codes of subexponential lentgh. *IEICE Transactions on Information and Systems*, pages 263–270, 2010.

[62] Stasys Jukna. *Extremal combinatorics.* 2001.

[63] Boris Kashin and Alexander Razborov. Improved lower bounds on the rigidity of hadamard matrices. *Mathematical Notes*, 63:471–475, 1998.

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

[65] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of Mersenne numbers. *SIAM Jounrnal on Computing*, 38:1952–1969, 2009.

[66] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69:395–420, 2004.

[67] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single-database computationally-private information retrieval. In *38rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 364–373, 1997.

[68] Rudolf Lidl and Harald Niederreiter. *Finite Fields.* 1983.

[69] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. International Association for Cryptologic Research Technical Report 2004/063, 2004.

[70] Richard Lipton. Efficient checking of computations. In *7th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 415 of Lecture Notes in Computer Science, pages 207–215, 1990.

[71] Satyanarayana Lokam. Spectral methods for matrix rigidity with applications to size-depth trade-offs and communication complexity. *Journal of Computer and System Sciences*, 63:449–473, 2001.

[72] Satyanarayana Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4:1–155, 2009.

[73] F.J. MacWilliams and N.J.A. Sloane. *The theory of error correcting codes.* 1977.

[74] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

[75] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *29th ACM Symposium on Theory of Computing (STOC)*, pages 245–254, 1999.

[76] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 369–377, 1999.

[77] Michael Nielsen and Isaac Chuang. *Quantum computation and quantum information.* 2000.

[78] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *6th International Workshop on Randomization and Computation (RANDOM)*, volume 2483 of Lecture Notes in Computer Science, pages 39–50, 2002.

[79] Rafail Ostrovsky and Victor Shoup. Private information storage. In *29th ACM Symposium on Theory of Computing (STOC)*, pages 294–303, 1997.

[80] Alexander Polishchuk and Daniel Spielman. Nearly-linear size holographic proofs. In *26th ACM Symposium on Theory of Computing (STOC)*, pages 194–203, 1994.

[81] Prasad Raghavendra. A note on Yekhanin's locally decodable codes. Electronic Colloquium on Computational Complexity (ECCC) TR07-016, 2007.

[82] Alexander Razborov and Sergey Yekhanin. An $\Omega(n^{1/3})$ lower bound for bilinear group based private information retrieval. *Theory of Computing*, 3:221–238, 2007.

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

[84] Andrei Romashchenko. Reliable computations based on locally decodable codes. In *23rd International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3884 of Lecture Notes in Computer Science, pages 537–548, 2006.

[85] Jiri Sgall. Bounds on pairs of families with restricted intersections. *Combinatorica*, 19:555–566, 1999.

[86] Amin Shokrollahi, Daniel Speilman, and Voelker Stemann. A remark on matrix rigidity. *Information Processing Letters*, 64:283–285, 1997.

[87] V. Shoup. *A computational introduction to number theory and algebra.* 2005.

[88] Madhu Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, University of California at Berkeley, 1992.

[89] Madhu Sudan. Personal communication, 2009.

[90] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. In *39th ACM Symposium on Theory of Computing (STOC)*, pages 537–546, 1999.

[91] Gabor Tardos and David Barrington. A lower bound of the mod 6 degree of the OR function. *Computational Complexity*, 7:99–108, 1998.

[92] Luca Trevisan. Coding theory and complexity. Lecture notes, 2003.

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

[94] Leslie Valiant. Graph-theoretic arguments in low level complexity. In *6th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 162–176, 1977.

[95] J.H. van Lint. *Introduction to Coding Theory.* 1982.

[96] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of Lecture Notes in Computer Science, pages 1424–1436, 2005.

[97] David Woodruff. New lower bounds for general locally decodable codes. Electronic Colloquium on Computational Complexity (ECCC) TR07-006, 2007.

[98] David Woodruff. Corruption and recovery-efficient locally decodable codes. In *International Workshop on Randomization and Computation (RANDOM)*, pages 584–595, 2008.

[99] David Woodruff. A quadratic lower bound for three query linear locally decodable codes over any field. In *International Workshop on Randomization and Computation (RANDOM)*, 2010.

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

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

[102] Sergey Yekhanin. Private information retrieval. *Communications of the ACM*, (4):68–73, 2010.