Codes with local decoding procedures

Sergey Yekhanin

Abstract. Error correcting codes allow senders to add redundancy to messages, encoding bit strings representing messages into longer bit strings called codewords, in a way that the message can still be recovered even if a fraction of the codeword bits are corrupted. In certain settings however the receiver might not be interested in recovering all the message, but rather seek to quickly recover just a few coordinates of it. Codes that allow one to recover individual message coordinates extremely fast (locally), from accessing just a small number of carefully chosen coordinates of a corrupted codeword are said to admit a local decoding procedure. Such codes have recently played an important role in several areas of theoretical computer science and have also been used in practice to provide reliability in large distributed storage systems. We survey what is known about these codes.

Mathematics Subject Classification (2010). Primary 94B05, 94B35; Secondary 68R05, 68P20.

Keywords. Error correcting codes, locally decodable codes, private information retrieval schemes, multiplicity codes, matching vectors codes, local reconstruction codes, maximally recoverable codes.

1. Introduction

The 60+ years of research in coding theory that started with the works of Shannon [24] and Hamming [14] gave us nearly optimal ways to add redundancy to messages, encoding bit strings representing messages 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 certain scenarios however, the receiver of the corrupted message might not be interested in recovering all the message, but rather seek to reconstruct just a small portion of it. For instance one can think of a setting where a large database is stored encoded with an error correcting code and a user who is willing to access a single database record. When classical codes are employed the user would have no alternative but to decode all the database (investing effort that is at least proportional to the database size) and then access the record. This example calls for codes that admit local decoding procedures, i.e., allow one to reliably recover individual message coordinates from accessing just a small number of coordinates of a corrupted codeword. The goal of our survey is to review the state of the art in such codes.

In what follows we model corrupted coordinates as being erased rather then flipped. This simplifies presentation and also allows us give a unified treatment of the few lines of work on the subject. We assume that the decoder is aware of which coordinates are missing. Below is a simple example of a code that admits a local decoding procedure.

Hadamard code. The code encodes k-bit messages $\mathbf{x}$ to 2^k-bit codewords $C(\mathbf{x})$. It allows any coordinate $\mathbf{x}(i)$ to be recovered by accessing just two coordinates of $C(\mathbf{x})$ even after almost a half of coordinates of $C(\mathbf{x})$ are erased. In what follows, let $[k]$ denote the set $\{1, \ldots, k\}$. Every coordinate of the Hadamard code corresponds to one (of 2^k) subsets of $[k]$ and stores the XOR of the corresponding bits of the message $\mathbf{x}$. Observe that for any set $S \subseteq [k]$, $\mathbf{x}(i)$ equals the XOR of the values stored at coordinates 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 less than half of coordinates of $C(\mathbf{x})$ are erased; then for every $i \in [k]$, there exists a set $S \subseteq [k]$ such that both coordinates corresponding to S and $S \triangle \{i\}$ are available, and thus $\mathbf{x}(i)$ can be recovered with two reads.

Codes with local decoding procedures vary in terms of the number of erasures after which local recovery can be guaranteed. The other two main parameters of interest 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 coordinates that need to be read from the corrupted codeword in order to recover a single coordinate of the message. For instance, in the Hadamard code above, local recovery is guaranteed after $2^{k-1} - 1$ erasures, redundancy equals $2^k - k$, and query complexity is 2.

In general one cannot optimize all three parameters discussed above simultaneously. There are tradeoffs. In this survey we restrict our attention to two main families of codes with local decoding procedures, namely, Locally Decodable Codes (LDCs) and Local Reconstruction Codes (LRCs). Locally decodable codes allow quick recovery of individual message coordinates in a very aggressive scenario when a linearly growing number of codeword coordinates might be missing. These codes play an important role in several areas of theoretical computer science and tend to require either a large amount of redundancy or a high query complexity. Local reconstruction codes, by contrast, only allow quick recovery when just a single coordinate is unavailable. As such they are considerably more efficient in terms of both codeword length and number of queries. Instances of these codes have been used in practice to provide reliability in large distributed storage systems.

Outline of the paper. In Sections 2 through 4 we deal with locally decodable codes. Section 2 provides a basic introduction to the area and discusses applications. Sections 3 and 4 cover the two main families of LDCs, namely multiplicity codes that are the most efficient codes in the regime of high query complexity and matching vector codes that are the best known codes in the regime of low query complexity. In Section 5 we review the state of the art in local reconstruction codes.

2. Basics of locally decodable codes

As we discussed above LDCs are erasure correcting codes that allow extremely efficient (sub-linear time) decoding of individual message coordinates even when a linearly growing number of codeword coordinates are unavailable. Below is a formal definition.

Definition 2.1. Let q be a prime power, $1 \leq r \leq k \leq N$ be integers, and $\delta > 0$ be real. Assume that for every $i \in [k]$ there is a collection $\mathcal{D}_i$ of r-subsets of $[N]$. An (r, δ)-locally decodable code is a mapping $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$ such that:

- For every message $\mathbf{x} \in \mathbb{F}_q^k$, for each $i \in [k]$ and $S \in \mathcal{D}_i$ the symbol $\mathbf{x}(i)$ can be recovered from accessing r coordinates of $C(\mathbf{x})$ that belong to S.
- For every set $E \subseteq [N]$ such that $|E| \leq \delta n$, for every $i \in [k]$ there exists a set $S \in \mathcal{D}_i$ such that $E \cap S = \emptyset$.

A locally decodable code is called *linear* if C is a linear transformation over $\mathbb{F}_q$. Almost all codes considered in the survey are linear.

Not all parameters of locally decodable codes are considered equally important. In what follows we will typically pay little attention to alphabet size q and fraction of erasures δ and focus on the values of the codeword length N and the query complexity r when the message length k grows to infinity. Ideally, one would like to have both N and r as small as possible. One however can not minimize the length and the query complexity simultaneously. There is a tradeoff. On one end of the spectrum we have classical error correcting codes 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.

2.1. Reed Muller codes.

In this section we discuss the oldest and most basic family of locally decodable codes. An LDC allows to quickly recover any coordinate of a message by accessing only few coordinates of its corrupted encoding. A related property is that of local correctability allowing to locally recover not only coordinates of the message but also arbitrary coordinates of the encoding.

Definition 2.2. Let q be a prime power, $1 \leq r \leq k \leq N$ be integers, and $\delta > 0$ be real. Assume that for every $i \in [N]$ there is a collection $\mathcal{D}_i$ of r-subsets of $[N]$. An (r, δ)-locally correctable code is a subset $C \subseteq \mathbb{F}_q^N$ of size q^k such that:

- For every codeword $\mathbf{x} \in C$, for each $i \in [N]$ and $S \in \mathcal{D}_i$ the symbol $\mathbf{x}(i)$ can be recovered from accessing r coordinates of $\mathbf{x}$ that belong to S.
- For every set $E \subseteq [N]$ such that $|E| \leq \delta n$, for every $i \in [N]$ there exists a set $S \in \mathcal{D}_i$ such that $E \cap S = \emptyset$.

We often refer to the quantity $\log_q |C|$ as the message length of a locally correctable code C. It is not hard to show that every linear locally correctable code yields a linear locally decodable code with the same parameters.

Reed Muller codes that we discuss below are locally correctable. In what follows a D-evaluation of a function h defined over a domain D, is a vector of values of h at all points of D. Also 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*. The key idea behind Reed Muller codes is that of polynomial interpolation. Messages are encoded by complete evaluations of low degree multivariate polynomials over a finite field. Local correctability is achieved through reliance on the rich structure of short local dependencies between such evaluations at multiple points.

A Reed Muller code is specified by three integer parameters, namely, a prime power (alphabet size) q, number of variables n, and a degree d. The q-ary code consists of $\mathbb{F}_q^n$-evaluations of all polynomials of total degree at most d in the ring $\mathbb{F}_q[z_1, \ldots, z_n]$. When viewed as an LDC such code encodes $k = \binom{n+d}{d}$-long messages over $\mathbb{F}_q$ to q^n-long codewords.

Below we present the simplest local corrector for Reed Muller codes. 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 picks an affine line through $\mathbf{w}$ and then relies on the local dependency between the values of F at any $d+2$ points along the line. Let $\mathbb{F}_q^*$ denote the multiplicative subgroup of the field $\mathbb{F}_q$.

Theorem 2.3. *Let n and d be positive integers. Let q be a prime power, $\delta > 0$ be a real, and $d < (1-\delta)q - 1$ be an integer; then there exists a linear code of dimension $k = \binom{n+d}{d}$ in $\mathbb{F}_q^N$, $N = q^n$, that is $(d+1, \delta)$-locally correctable.*

Proof. The code consists of $\mathbb{F}_q^n$-evaluations of all polynomials of total degree at most d in the ring $\mathbb{F}_q[z_1, \ldots, z_n]$. Consider the i-th coordinate, $i \in [N]$ corresponding to a point $\mathbf{w} \in \mathbb{F}_q^n$. The family $\mathcal{D}_i$ consists of all $(d+1)$-tuples of points that can be obtained by picking a nontrivial affine line

$$L = \{\mathbf{w} + \lambda \mathbf{v} \mid \lambda \in \mathbb{F}_q^*\} \tag{1}$$

through $\mathbf{w}$ and fixing some $d+1$ points on it. The local correction procedure is quite natural. The decoder reads the values of the polynomial F at $d+1$ points of some undamaged set $S \in \mathcal{D}_i$. Note that such a set always exists under the assumptions of the theorem. Assume the set S comes from line (1). The decoder invokes univariate polynomial interpolation to recover the degree d polynomial f which is the restriction of F to the line L, i.e., $f(\lambda) = F(\mathbf{w} + \lambda \mathbf{v})$. The decoder outputs $f(0) = F(\mathbf{w})$. □

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 and letting these parameters grow to infinity. Increasing d relative to n yields shorter codes of larger query complexity. Below we summarize asymptotic

parameters of several families of locally decodable codes based on Reed Muller codes.

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

2.2. Applications. Interestingly, the natural application of locally decodable codes to data storage mentioned in Section 1 is neither the historically earliest nor the most important. LDCs have a host of applications in other areas of theoretical computer science such as complexity theory, data structures, and derandomization. However their most prominent application is in cryptography to the design of Private Information Retrieval schemes (PIRs). In what follows we briefly review this application.

Private information retrieval schemes are cryptographic protocols designed to safeguard the privacy of database users. They allow clients to retrieve records from replicated public databases while completely hiding the identity of the retrieved records from database owners. 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's interest. Below we demonstrate a general procedure that obtains an r-server PIR scheme out of any r-query *smooth* LDC. A locally decodable code is called smooth if for every $i \in [k]$, the r-tuples in $\mathcal{D}_i$ cover the universe $[N]$ uniformly, i.e., each $j \in [N]$ belongs to the same number of sets in $\mathcal{D}_i$. Almost all known constructions of LDCs yield smooth codes.

Let C be a smooth LDC encoding k-bit messages to N-bit codewords. At the preprocessing stage servers $\mathcal{S}_1, \ldots, \mathcal{S}_r$ encode the k-bit database $\mathbf{x}$ with the code C. Next the user $\mathcal{U}$ who is interested in obtaining the i-th bit of $\mathbf{x}$, picks an r-tuple of queries $(\mathrm{que}_1, \ldots, \mathrm{que}_r) \in \mathcal{D}_i$ uniformly at random. 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 one bit answer $C(\mathbf{x})_{\mathrm{que}_j}$, which is the value of $C(\mathbf{x})$ at coordinate que_j. The user combines servers' responses to obtain $\mathbf{x}(i)$.

It is straightforward to verify that the protocol above is private since by the smoothness property for every $j \in [r]$ the query que_j is uniformly distributed over the set of codeword coordinates. The total communication is given by

$$r(\log N + 1).$$

Thus short codes of low query complexity yield communication efficient PIR schemes involving a small number of servers. For example, instantiating the reduction above with the best known 3-query LDCs yields 3-server private information retrieval schemes with $O\left(2^{\sqrt{\log k \log\log k}}\right)$ communication to access a k-bit database.

2.3. Notes. Formal definition of locally decodable codes was given in 2000 by Katz and Trevisan [18], who cited Leonid Levin for inspiration. See also [25]. However codes that meet this definition have been around for much longer. The

first such family of codes, namely, Reed Muller codes [21, 23], were introduced in 1950s, and their local decodability properties have been exploited implicitly since then. Today there are few families of locally decodable codes that surpass Reed Muller codes in terms of query complexity vs. codeword length tradeoff. We are going to discuss two of them (namely, multiplicity codes [20] and matching vector codes [28, 9]) in the following sections. For a detailed survey of the locally decodable codes see [29]. Private Information Retrieval (PIR) schemes were introduced in [4]. See [27] for a recent survey.

3. Multiplicty codes

When dealing with codes that have low redundancy it is convenient to utilize the notion of code rate. For an (r, δ)-locally decodable code $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$, the rate k/N is simply the ratio of the number of message symbols to the number of codeword symbols. Similarly, for a (r, δ)-locally correctable code $C \subseteq \mathbb{F}_q^N$, the rate is the ratio $(\log_q |C|)/N$. In applications to data transmission and storage one is naturally interested in codes of high rate, i.e., rate close to 1.

In this section we review multiplicity codes. These codes generalize Reed Muller codes (Theorem 2.3) and improve upon them in the high rate regime. Observe that with Reed Muller codes the rate can never be too high. Recall that these codes are specified by three parameters: alphabet size q, total degree d, and the number of variables n. The rate is highest when $n = 2$, $d = (1 - \delta)q$, and q grows to infinity. In this setting we have $k = \binom{d+2}{2}$ and $N = q^2$, thus the rate is $\binom{d+2}{2}/q^2 \approx \frac{1}{2}$ and query complexity $r = O(\sqrt{k})$.

Note that with Reed Muller codes, one cannot increase the rate by simply allowing evaluations of higher degree polynomials, as if one allows the degree to exceed the field size, one starts getting polynomials with colliding evaluations such as z and z^q. Multiplicity codes, however, use much higher degree polynomials and thus have significantly improved rates, and avoid the pitfall mentioned above by evaluating polynomials *together with their partial derivatives*.

In what follows we review the construction of multiplicity codes. We consider the simplest example of these codes based on bivariate polynomials, which have improved rate above $\frac{1}{2}$, and see how to locally correct them with essentially the same query complexity $O(\sqrt{k})$. Finally, we mention how general multiplicity codes are defined and discuss some of the ideas that go into locally correcting them. Our main result gives codes that simultaneously have rate approaching one, and allow for local correction with arbitrary polynomially-small number of queries.

3.1. Bivariate multiplicity codes. Let q be a prime power, let $\delta > 0$ and let integer $d = 2(1 - \delta)q$. The multiplicity code of *order two* evaluations of degree d bivariate polynomials over $\mathbb{F}_q$ is the code defined as follows. As before, the coordinates are indexed by $\mathbb{F}_q^2$ (so $N = q^2$) and the codewords are indexed by bivariate polynomials of degree at most d over $\mathbb{F}_q$. However the alphabet now is

$\mathbb{F}_q^3$. The codeword corresponding the polynomial $F(x_1, x_2)$ is the vector

$$C(F) = \left\langle \left(F(\mathbf{w}), \frac{\partial F}{\partial x_1}(\mathbf{w}), \frac{\partial F}{\partial x_2}(\mathbf{w}) \right) \right\rangle_{\mathbf{w} \in \mathbb{F}_q^2} \in \left(\mathbb{F}_q^3\right)^{q^2}.$$

In words, the $\mathbf{w}$ coordinate consists of the evaluation of F and its formal partial derivatives $\frac{\partial F}{\partial x_1}$ and $\frac{\partial F}{\partial x_2}$ at $\mathbf{w}$. Because two distinct polynomials of degree at most d can agree with multiplicity two on at most $d/2q$-fraction of the points in $\mathbb{F}_q^2$ no two codewords defined above collide. Since the alphabet size is now q^3, the rate of the new code is

$$\frac{\log_{q^3} q^{\binom{d+2}{2}}}{q^2} = \frac{\binom{d+2}{2}}{3q^2} \approx \frac{2}{3}(1-\delta)^2.$$

Summarizing the differences between this multiplicity code with the Reed Muller code described earlier:

- Instead of polynomials of degree $(1-\delta)q$, we consider polynomials of degree double of that.
- Instead of evaluating the polynomials, we take their *order two* evaluation.

This yields a code with the rate improved from below $1/2$ to nearly $2/3$. We now argue that the new code is still locally correctable with $O(\sqrt{k})$ queries.

Local correction of multiplicity codes: Given the codeword corresponding to the polynomial $F(x_1, x_2)$ with some (say, $\delta/2$) fraction of coordinates erased and given a point $\mathbf{w} \in \mathbb{F}_q^2$, we want to recover the symbol at coordinate $\mathbf{w}$, namely

$$\left(F(\mathbf{w}), \frac{\partial F}{\partial x_1}(\mathbf{w}), \frac{\partial F}{\partial x_2}(\mathbf{w}) \right). \tag{2}$$

Similarly to the case of Reed Muller codes, the algorithm picks a direction $\mathbf{v} \in \mathbb{F}_q^2$ such that less than a δ fraction of coordinates in the affine line

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

are missing. Most directions are like this. Our intermediate goal is to recover the univariate polynomial $f(\lambda) = F(\mathbf{w} + \lambda \mathbf{v})$. The important observation here is that for every $\lambda_0 \in \mathbb{F}_q$, the $\mathbf{w} + \lambda_0 \mathbf{v}$ coordinate of $C(F)$ completely determines both the value and the first derivative of the univariate polynomial $f(\lambda)$ at the point λ_0. Indeed,

$$\begin{aligned} f(\lambda_0) &= F(\mathbf{w} + \mathbf{v}\lambda_0), \\ \frac{\partial f}{\partial \lambda}(\lambda_0) &= \frac{\partial F}{\partial x_1}(\mathbf{w} + \mathbf{v}\lambda_0) \cdot \mathbf{v}(1) + \frac{\partial F}{\partial x_2}(\mathbf{w} + \mathbf{v}\lambda_0) \cdot \mathbf{v}(2), \end{aligned}$$

where the last identity follows by the chain rule. Thus our knowledge of $C(F)$ at $(1-\delta)q + 1$ locations on the line L gives us access to $(1-\delta)q + 1$ evaluations of the polynomial $f(\lambda)$ and its derivative $\frac{\partial f}{\partial \lambda}(\lambda_0)$, where $f(\lambda)$ is of degree $\leq 2(1-\delta)q$.

This is enough to recover the polynomial $f(\lambda)$. Evaluating $f(\lambda)$ at $\lambda = 0$ gives us $F(\mathbf{w})$. Evaluating the derivative $\frac{\partial f}{\partial \lambda}(\lambda)$ at $\lambda = 0$ gives us the directional derivative of F at $\mathbf{w}$ in the direction $\mathbf{v}$ (which equals $\frac{\partial F}{\partial x_1}(\mathbf{w}) \cdot \mathbf{v}(1) + \frac{\partial F}{\partial x_2}(\mathbf{w}) \cdot \mathbf{v}(2)$).

We have clearly progressed towards our goal of computing $C(F)_{\mathbf{w}}$ given by formula (2), but we are not yet there. The final observation is that if we pick another direction $\mathbf{v}'$, that is not collinear with $\mathbf{v}$ and repeat the above process to recover the directional derivative of F at $\mathbf{w}$ in direction $\mathbf{v}'$, then the two directional derivatives of F at $\mathbf{w}$ in directions $\mathbf{v}, \mathbf{v}'$ together suffice to recover $\frac{\partial F}{\partial x_1}(\mathbf{w})$ and $\frac{\partial F}{\partial x_2}(\mathbf{w})$, as desired. This algorithm makes less than $2q$ queries, which is $O(\sqrt{k})$.

3.2. General multiplicity codes. The basic example of a multiplicity code above already achieves rate above 1/2. To get codes of rate approaching 1, one needs to modify the construction by considering evaluations of all derivatives of F up to an even higher order. In order to locally recover the higher-order derivatives of F at a point $\mathbf{w}$, the decoding algorithm picks many lines passing through $\mathbf{w}$, recovers the restriction of F to those lines, and combines all these recovered univariate polynomials in a certain way.

To reduce the query complexity to $O(k^{\epsilon})$ for small ϵ, one needs to modify the above example by considering multivariate polynomials in a larger number of variables n. The local decoding algorithm for this case, in order to locally recover at a point $\mathbf{w} \in \mathbb{F}_q^n$, still decodes by picking lines passing through $\mathbf{w}$; the reduced query complexity occurs because lines (with only q points) are now much smaller relative to a higher dimensional space $\mathbb{F}_q^n$.

Increasing both the maximum order of derivatives taken and the number of variables simultaneously yields multiplicity codes with rate close to one and arbitrarily low polynomial query complexity.

Theorem 3.1. *Let q be an arbitrary prime power. For every real $\epsilon, \alpha > 0$ there exists a real $\delta > 0$ such that for all sufficiently large message lengths k, there exists an $\mathbb{F}_q$-linear $(O(k^{\epsilon}), \delta)$-locally correctable code of rate $1 - \alpha$.*

3.3. Notes. Multiplicity codes were introduced in [20]. The construction builds on some technical tools from [7]. Alternative constructions with similar parameters have been given in [13, 15]. It is plausible that the query complexity of locally decodable codes of rate close to one can be further reduced. The only available lower bound is $\Omega(\log k)$ from [18].

4. Matching vector codes

In this section we review locally decodable codes that arise from families of matching vectors. Matching Vector (MV) codes are important as they exhibit dramatically better parameters than Reed Muller codes in the regime of low query complexity. Any construction of such codes naturally falls into two parts: the design of a matching vector family, and the actual code construction. Our focus is on the

second part.

4.1. The framework. MV codes inherit some structure from Reed Muller codes. A matching vector 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 local decoders for Reed Muller codes. It operates by picking a line in a certain direction and decoding 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. Also, let $\mathbf{u} \cdot \mathbf{v}$ denote the usual dot product of vectors $\mathbf{u}$ and $\mathbf{v}$.

Definition 4.1. 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 $i \in [k]$, $\mathbf{u}_i \cdot \mathbf{v}_i = 0$;
- For all $i, j \in [k]$ such that $i \neq j$, $\mathbf{u}_j \cdot \mathbf{v}_i \in S$.

We now show how one can obtain a 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

$$M_{\mathbf{w},\mathbf{v}} = \left\{ g^{\mathbf{w}+\lambda\mathbf{v}} \mid \lambda \in \mathbb{Z}_m \right\}; \tag{3}$$

- For $\mathbf{u} \in \mathbb{Z}_m^n$, we define the monomial

$$\mathrm{mon}_{\mathbf{u}} \in \mathbb{F}_q[z_1, \ldots, z_n]/(z_1^m = 1, \ldots, z_n^m = 1)$$

by

$$\mathrm{mon}_{\mathbf{u}}(z_1, \ldots, z_n) = \prod_{\ell \in [n]} z_\ell^{\mathbf{u}(\ell)}. \tag{4}$$

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

$$\mathrm{mon}_{\mathbf{u}}\left(g^{\mathbf{w}+\lambda\mathbf{v}}\right) = g^{\mathbf{u}\cdot\mathbf{w}} \left(g^{\lambda}\right)^{\mathbf{u}\cdot\mathbf{v}}. \tag{5}$$

This suggests that if we set $y = g^{\lambda} \in \mathbb{F}_q^*$ in formula (5); then what we get is a univariate polynomial in y. Hence the $M_{\mathbf{w},\mathbf{v}}$-evaluation of a monomial $\mathrm{mon}_{\mathbf{u}}$ is a $\mathbb{C}_m$-evaluation of a univariate monomial

$$g^{\mathbf{u}\cdot\mathbf{w}} y^{\mathbf{u}\cdot\mathbf{v}} \in \mathbb{F}_q[y]. \tag{6}$$

This observation is the foundation of all decoding algorithms for MV codes.

We now specify the encoding procedure and the most basic decoding procedure. Let $\mathcal{U}, \mathcal{V}$ be an S-matching family in $\mathbb{Z}_m^n$, where $|\mathcal{U}| = |\mathcal{V}| = k$.

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

$$F(z_1, \ldots, z_n) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot \mathrm{mon}_{\mathbf{u}_j}(z_1, \ldots, z_n). \tag{7}$$

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).

Decoding: The input to the decoder is an $\mathbb{C}_m^n$-evaluation of F with some δ fraction of coordinates erased and an index $i \in [k]$.

1. The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ such that none of the values of F at points of the the multiplicative line $M_{\mathbf{w},\mathbf{v}_i}$ are erased. If $\delta < \frac{1}{m}$ such a $\mathbf{w}$ exists.

2. The decoder recovers the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$. To accomplish this the decoder queries the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at $|S| + 1$ locations

$$\left\{ g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \{0, \ldots, s\} \right\}. \tag{8}$$

Firstly, let us see how the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F uniquely determines $\mathbf{x}(i)$. Observe that by formulas (5), (6) and (7) the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is the $\mathbb{C}_m$-evaluation of a polynomial

$$f(y) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot g^{\mathbf{u}_j\cdot\mathbf{w}} y^{\mathbf{u}_j\cdot\mathbf{v}_i} \in \mathbb{F}_q[y]. \tag{9}$$

Properties of the S-matching family $\mathcal{U}, \mathcal{V}$ imply that $y^{(\mathbf{u}_j,\mathbf{v}_i)} = 1$, if $j = i$; and $y^{\mathbf{u}_j\cdot\mathbf{v}_i} = y^s$, for some $s \in S$ otherwise. Formula (9) yields

$$f(y) = \mathbf{x}(i) \cdot g^{\mathbf{u}_i\cdot\mathbf{w}} + \sum_{s\in S} \left(\sum_{j \,:\, \mathbf{u}_j\cdot\mathbf{v}_i = s} \mathbf{x}(j) \cdot g^{\mathbf{u}_j\cdot\mathbf{w}} \right) y^s. \tag{10}$$

For a polynomial $h \in \mathbb{F}_q[y]$ we denote by $\mathrm{supp}(h)$ 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 (10) that $\mathrm{supp}(f) \subseteq S \cup \{0\}$ and

$$\mathbf{x}(i) = f(0)/g^{\mathbf{u}_i\cdot\mathbf{w}}. \tag{11}$$

Secondly, let us note that recovering the polynomial (10) from the values $\{c_0, \ldots, c_s\}$ of F at locations locations (8) is quite straightforward. The decoder simply recovers the unique sparse univariate polynomial $h(y) \in \mathbb{F}_q[y]$ with $\mathrm{supp}(h) \subseteq S \cup \{0\}$ such that for all $\lambda \in \{0, \ldots, s\}$, $h(g^\lambda) = c_\lambda$. The uniqueness of such $h(y) = f(y)$ follows from standard properties of Vandermonde matrices.

Putting it all together we obtain the following

Proposition 4.2. *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 $\mathbb{F}_q$-linear code encoding k-long messages to m^n-long codewords that is $(s+1, \frac{1}{m})$-locally decodable.*

As Proposition 4.2 suggests parameters of matching vector codes are governed by parameters of the underlying family of matching vectors. To get short codes of low query complexity we need large S-matching families for small sets S. The best constructions of such families are given by the following

Proposition 4.3. *Let $m = p_1 \ldots p_t$ be a product of t distinct primes. There exists a set $S \subseteq \mathbb{Z}_m \setminus \{0\}$, $|S| = 2^t - 1$ such that for all sufficiently large integers n, there is an S-matching family in $\mathbb{Z}_m^n$ of size*

$$n^{c\left(\frac{\log n}{\log \log n}\right)^{t-1}},$$

where the constant c depends only on m.

Combing the two propositions above we get

Theorem 4.4. *Let $m = p_1 \ldots p_t$ be a product of t distinct primes. Let q be a prime power such that $m \mid q-1$; then for infinitely many values of message length k there exists an $\mathbb{F}_q$-linear $(2^t, \frac{1}{m})$-locally decodable code of codeword length*

$$N = \exp\exp\left(O(\sqrt[t]{\log k (\log\log k)^{t-1}})\right).$$

Observe that for constant t the function above grows slower than any exponential function of the form $2^{\alpha k}$ though faster than any polynomial k^c.

4.2. Notes. Constructions of locally decodable codes from matching vectors originated in [28] and were developed further in [9, 6, 2]. An important progress in this line of work has been accomplished by Klim Efremenko in [9] where the first constructions of codes from matching vectors modulo composites (rather than primes) were considered. Proposition 4.3 is due to Grolmusz [12]. An important ingredient to his proof is the low-degree representation of the OR-function from [1].

Despite considerable progress in constructions of locally decodable codes of small query complexity we are still very far from closing the gap to lower bounds. It is only in the setting of 2-query codes that we know the true codeword length of optimal LDC, which is exponential [19]. For any other number of queries large gaps remain. For instance, in the case of the three-query codes the best upper bound for the codeword length comes from matching vector codes and is

$\exp\exp(\sqrt{\log k \log\log k})$ while the best lower bound is $\Omega(k^2)$ from [26]. Closing this gap is a major open problem.

Locally decodable codes are also of interest over infinite fields. Questions about these codes relate to classical problems in combinatorial geometry [5].

5. Local reconstruction codes

In previous sections we reviewed the state of the art in locally decodable codes, i.e., codes that admit local recovery of individual message symbols in the regime when a linearly growing number of codeword coordinates may be unavailable. As we saw these codes are either highly redundant or have a large query complexity. In particular, to get rate close to one the best known LDCs need polynomially many queries, and to get constant query complexity independent of the message length they need super-polynomial codeword length.

In the current section we turn our attention to local reconstruction codes which only allow local recovery when just a single coordinate is unavailable, while also providing non-local recovery guarantees after a larger number of erasures. Since LRCs are geared towards less aggressive failure scenarios than LDCs they are considerably simpler and more efficient. We start be reviewing the motivation behind these codes.

5.1. Applications. Modern large scale distributed storage systems such as data centers store data in a redundant form to ensure reliability against node (e.g., individual machine) failures. The simplest solution here is the straightforward replication of data packets across different nodes. Alternative solution involves erasure coding: the data is partitioned into k information packets. Subsequently, using an erasure code, $N-k$ parity packets are generated and all N packets are stored in different nodes.

Using erasures codes instead of replication may lead to dramatic improvements both in terms of redundancy and reliability. However to realize these improvements one has to address the challenge of maintaining an erasure encoded representation. In particular, when a node storing some packet fails, one has to be able to quickly reconstruct the lost packet in order to keep the data readily available for the users and to maintain the same level of redundancy in the system. We say that a certain packet has *locality* r if it can be recovered from accessing only r other packets. One way to ensure fast reconstruction is to use erasure codes where all packets have low locality $r \ll k$. Having small value of locality is particularly important for information packets.

These considerations lead to introduction of (r,d)-local reconstruction codes, i.e., a linear codes capable of correcting any $d-1$ erasures where all information symbols have locality at most r. Storage systems based on (r,d)-codes provide fast recovery of information packets from a single node failure (typical scenario), and ensure that no data is lost even if up to $d-1$ nodes fail simultaneously.

5.2. Structure of LRCs. We begin by introducing some basic notions. A linear code is a linear mapping $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$, where $k \leq N$. Every such mapping can be represented as

$$C(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{p}_1, \ldots, \mathbf{x} \cdot \mathbf{p}_N), \tag{12}$$

where $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{F}_q^k$. We say that C is a systematic code if all for all $i \in [k]$, $\mathbf{p}_i$ is the i-th unit vector, i.e., the vector whose unique non-zero coordinate i carries value 1. In other words, a code is systematic if it performs encoding by appending redundant symbols to the original message. We refer to coordinates 1 through k of a systematic code as information coordinates.

We say that the i-th coordinate of C has locality r if, when erased, this coordinate can be recovered by accesing at most r of the $N-1$ remaining coordinates of a codeword. This is equivalent to saying that the vector $\mathbf{p}_i$ in (12) is in the span of some r vectors of $\{\mathbf{p}_j\}_{j \in [N] \setminus \{i\}}$. Further we say that a systematic code C has information locality r, if all information coordinates of C have locality r. Finally we say that a code C has distance d, if C corrects any pattern of up to $d-1$ simultaneous erasures.

Definition 5.1. A linear systematic code $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$ that has distance d and information locality r is called an (r, d)-local reconstruction code.

Below we present one simple family of (r, d)-local reconstruction codes, called Pyramid codes. We assume $r \mid k$.

Pyramid codes. To define an (r, d)-Pyramid code C encoding messages of dimension k we fix an arbitrary linear systematic code $E : \mathbb{F}_q^k \to \mathbb{F}_q^N$ that has distance d and codeword length $N = k - d + 1$. Note that such a code always exist provided that $q \geq N - 1$. Let

$$E(\mathbf{x}) = (\mathbf{x}, \mathbf{p}_0 \cdot \mathbf{x}, \mathbf{p}_1 \cdot \mathbf{x}, \ldots, \mathbf{p}_{d-2} \cdot \mathbf{x}).$$

We partition the set $[k]$ into $t = \frac{k}{r}$ disjoint subsets of size r, $[k] = \bigsqcup_{j \in [t]} S_j$. For a k-dimensional vector $\mathbf{x}$ and a set $S \subseteq [k]$ let $\mathbf{x}|_S$ denote the $|S|$-dimensional restriction of $\mathbf{x}$ to coordinates in the set S. We define the systematic code C by

$$C(\mathbf{x}) = (\mathbf{x}, (\mathbf{p}_0|_{S_1} \cdot \mathbf{x}|_{S_1}), \ldots, (\mathbf{p}_0|_{S_t} \cdot \mathbf{x}|_{S_t}), \ \mathbf{p}_1 \cdot \mathbf{x}, \ldots, \mathbf{p}_{d-2} \cdot \mathbf{x}.)$$

It is not hard to verify that the code C has distance d. We now argue that each information symbol $i \in [k]$ has locality r. Consider an arbitrary $i \in S_j$. Note that the value of $\mathbf{x}(i)$ can be deduced from accessing the light parity $\mathbf{p}_0|_{S_j} \cdot \mathbf{x}|_{S_j}$ and the values of information symbols $\mathbf{x}(l)$ for $l \in S_j \setminus \{i\}$.

Interestingly the simple construction above yields (r, d)-LRCs of the lowest possible redundancy.

Theorem 5.2. *For any linear code $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$ of distance d and information locality r,*

$$N \geq k + \left\lceil \frac{k}{r} \right\rceil + d - 2. \tag{13}$$

5.3. Maximal recoverability. A stronger version of Theorem 5.2 shows that under some minor technical assumptions all (r, d)-LRCs of the lowest possible redundancy are in a certain sense very similar to Pyramid codes. In particular, such codes have the same *topology*, i.e., the same set of dependency relations between information symbols and parity symbols. Specifically, assuming $r \mid k$:

- Data symbols are partitioned into k/r groups of size r. For each such group there is one (local) parity symbol that stores the XOR (or some other non-trivial linear combination) of respective data symbols.
- The remaining $h = d-2$ (heavy) parity symbols depend on all k data symbols.

In what follows we refer to codes meeting the description above as data-local (k, r, h)-codes. We also refer to a group of r data symbols and their local parity as a local group. (r, d)-LRCs with optimal redundancy are instances of data-local (k, r, h)-codes with $h = d - 2$.

Note that the class data-local codes is fairly broad as there is a lot of flexibility is choosing coefficients in heavy parities. All these codes have appropriate information locality. However they differ in terms of reliability guarantees that they provide as correctability of a particular failure pattern obviously depends on coefficients used to define heavy parities.

We say that a data-local (k, r, h)-code is Maximally Recoverable (MR) if it corrects every failure patterns that is correctable by some other code that has the same topology. Another equivalent way to define maximally recoverability is as follows:

Definition 5.3. Let C be a data-local (k, r, h)-code. We say that C is maximally recoverable if it corrects any failure pattern that can be obtained by erasing one coordinate in each of $\frac{k}{r}$ local groups as well as h arbitrary additional coordinates.

Note that maximal recoverability is a much stronger property than the mere distance required in the definition of (r, d)-local reconstruction codes.

It is not hard to show that maximally recoverable codes exist. In fact, simply picking coefficients of heavy parities at random from a large enough finite field with high probability yields an MR code. However in applications we would like to have codes over small finite fields to facilitate fast encoding and decoding. The best explicit constructions of such codes are given by the following

Theorem 5.4. *For constants r and h and for all k such that $r \mid k$, there exists a maximally recoverable data-local (k, r, h)-code over a field of size $O\left(k^{\lceil (h-1)\left(1-\frac{1}{2^r}\right)\rceil}\right)$.*

5.4. Notes. Pyramid codes were introduced in [16]. General local reconstruction codes were studied in [10]. Theorem 5.4 is from [11]. See also [3]. Local reconstruction codes are used in practice. Instances of these codes were first deployed by Windows Azure Storage [17], and have later been used in a number of other production systems. A different other notion of local reconstruction in codes for storage has been addressed in [8].

The main open challenge in the area of local reconstruction codes is to reduce the field size of maximally recoverable codes. The best upper bound for the field size is roughly $O(k^{h-1})$ while the only available lower bound is $\Omega(k)$ independent of h. Constructing explicit maximally recoverable codes over small finite fields in more general topologies is also of great interest.

6. Conclusion

In this survey we reviewed two main families of codes with local decoding procedures, namely locally decodable codes and local reconstructions codes. There is a large array of questions that remain open. In the case of LDCs the main open questions pertain to the true shape of the tradeoff between codeword length and query complexity. In the case of LRCs this tradeoff is understood and the main challenges are in constructing maximally recoverable codes over small finite fields. There is also a large area dealing with codes that provide local recovery of message symbols after more than one but less than $\Omega(N)$ erasures and thus bridge LDCs and LRCs. While there are some well studied families of codes that fall in this range, e.g., projective geometry codes [22], in general this regime is not well understood.

References

[1] Barrington, D. A., Beigel, R., and Rudich, S., Representing Boolean functions as polynomials modulo composite numbers, *Computational Complexity* **4** (1994), 367-382.

[2] Ben-Aroya, A., Efremenko, K., and Ta-Shma, A., Local list decoding with a constant number of queries. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science (FOCS)* (2010), 715-722.

[3] Blaum, M., Hafner, J. L., and Hetzler, S., Partial-MDS codes and their application to RAID type of architectures, *IEEE Transactions on Information Theory* **59** (7) (2013), 4510-4519.

[4] Chor, B., Goldreich, O., Kushilevitz., E., and Sudan, M., Private information retrieval, *Journal of the ACM* **45** (1998), 965-981.

[5] Dvir, Z., Incidence theorems and their applications, *Foundations and trends in theoretical computer science* **6** (4) (2012), 257-393.

[6] Dvir, Z., Gopalan, P., and Yekhanin, S., Matching vector codes, *SIAM Journal on Computing* **40** (4) (2011), 1154-1178.

[7] Dvir, Z., Kopparty, S., Saraf., S., and Sudan, M., Extensions to the method of multiplicities, with applications to Kakeya sets and mergers, *SIAM Journal on Computing* **42** (6) (2013), 2305-2328.

[8] Dimakis, A. G., Godfrey, B., Wu, Y., Wainwright, M. J., and Ramchandran, K., Network coding for distributed storage systems, *IEEE Transactions on Information Theory* **56** (2010), 4539-4551.

[9] Efremenko, K., 3-query locally decodable codes of subexponential length, *SIAM Journal on Computing* **41** (6) (2012), 1694-1703.

[10] Gopalan, P., Huang, C., Simitci, H., and Yekhanin, S., On the locality of codeword symbols, *IEEE Transactions on Information Theory* **58** (11) (2012), 6925-6934.

[11] Gopalan, P., Huang, C., Jenkins, B., and Yekhanin, S., Explicit maximally recoverable codes with locality, *Proceedings of the Electronic Colloquium on Computational Complexity (ECCC)* **20** (2013).

[12] Grolmusz, V., Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs, *Combinatorica* **20** (2000), 71-86.

[13] Guo, A., Kopparty, S., and Sudan, M., New affine-invariant codes from lifting., In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)* (2013), 529-540.

[14] Hamming, R. W., Error detecting and error correcting codes, *The Bell System Technical Journal* **26** (2) (1950), pp. 147-160.

[15] Hemenway, B., Ostrovsky, R., and Wootters, M., Local correctibility of expander codes, In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)* (2013), 540-551.

[16] Huang, C., Chen, M., and Li, J., Pyramid Codes: flexible schemes to trade space for access efficiency in reliable data storage systems, In *Proceedings of 6th IEEE International Symposium on Network Computing and Applications (NCA)* (2007), 79-86.

[17] Huang, C., Simitci, H., Xu, Y., Ogus, A., Calder, B., Gopalan, P., Li, J., and Yekhanin, S., Erasure coding in Windows Azure Storage, In *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)* (2012), 15-27.

[18] Katz, J. and Trevisan, L., On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)* (2000), 80-86.

[19] Kerenidis, I., and de Wolf, R., Exponential lower bound for 2-query locally decodable codes via a quantum argument, *Journal of Computer and System Sciences* **69** (2004), 395-420.

[20] Kopparty, S., Saraf, S., and Yekhanin, S., High-rate codes with sublinear-time decoding. In *Proceedings of the 43nd ACM Symposium on Theory of Computing (STOC)* (2011), 176-176.

[21] Muller, D. E., Application of boolean algebra to switching circuit design and to error detection, *IEEE Transactions on Computers* **3** (1954), 6-12.

[22] Peterson, W. W., and Weldon, E. J., *Error correcting codes*. Princeton Mathematical Series 39, The MIT Press, Cambridge, MA, 1972.

[23] Reed, I. S., A class of multiple-error-correcting codes and the decoding scheme, *IEEE Transactions on Information Theory* **4** (1954), 38-49.

[24] Shannon, C. E., A mathematical theory of communication, *The Bell System Technical Journal* **27** (1948), pp. 379-423, 623-656.

[25] Sudan, M., Trevisan, L., and Vadhan, S., Pseudorandom generators without the XOR lemma. In *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC)* (1999), 537-546.

[26] Woodruff, D., A quadratic lower bound for three query linear locally decodable codes over any field. In *Proceedings of the International Workshop on Randomization and Computation (RANDOM)* (2010), 766-779.

[27] Yekhanin, S., Private information retrieval, *Communications of the ACM* **53** (4) (2010), 68-73.

[28] Yekhanin, S., Towards 3-query locally decodable codes of subexponential length, *Journal of the ACM* **55** (2008), 1-16.

[29] Yekhanin, S., Locally decodable codes, *Foundations and trends in theoretical computer science* **6** (3) (2012), 139-255.

Microsoft Research, 1065 La Avenida, Mountain View, 94043, USA.

E-mail: yekhanin@microsoft.com