Locally decodable codes

Ehsan Ebrahimi Targhi
University of Tartu

Abstract. Locally decodable codes (LDCs) are error correcting codes that simultaneously provide efficient random-access to encoded data and high noise resilience by allowing reliable reconstruction of an arbitrary data bit from looking at only a small number of randomly chosen codeword bits. In this work we survey three known families of LDCs and compare their parameters.

1 Introduction

Locally decodable codes are a special family of error-correcting codes. The classical error-correcting codes allow one to encode a k-bit message into N-bit codeword $C(x)$, in such a way that x can still be recovered even if $C(x)$ gets corrupted in a number of coordinates. The traditional way to recover information about x given access to a corrupted version of $C(x)$ is to run a decoder for C, which would read and process the entire corrupted codeword, and then recover the entire original message x. Suppose that one is only interested in recovering a single bit or a few bits of x. In that case, LDCs have more efficient decoding schemes, while allowing one to read only a small number of code positions.

Additionally to k and N, the parameters of a locally decodable code are r, δ, ϵ. In what follows, we explain those parameters. The code rate is defined as a ratio k/N. Having a codeword with high rate is equivalent to have a codeword with low redundancy, where the redundancy is the number of bits used to transmit a message minus the number of bits of actual information in the message or $N - k$. The parameter ϵ is an upper bound on the probability that the decoder will not succeed to recover the corresponding bit correctly. The parameter δ is fraction of errors in a corrupted codeword and the parameter r is the number of queries that the local decoder needs to query from a (corrupted) codeword in order to recover the required bit. Ideally, one is interested to achieve the best trade-off between those parameters. The exact asymptotic trade-offs between the code parameters are not known. Finding such a trade-offs is still a major open problem in theoretical computer science.

We can name private information retrieval, secure multiparty computation, and average case complexity as some applications of LDCs in cryptography and computational complexity theory [2].

2 Preliminaries

Below, we introduce notations that are used in the paper:

Notations	
k/N	code rate
r	number of queries
δ	fraction of errors
ϵ	upper bound on the probability that the decoder will not succeed
$[k]$	$\{1, \ldots, k\}$
$\mathbb{F}_q$	finite field of q elements
$\mathbb{F}_q^*$	the multiplicative group of $\mathbb{F}_q$
(x, y)	scalar product of vectors x and y
$d(x, y)$	number of coordinates where x and y differ(Hamming distance).
$w(l)$	l-th coordinate of w when $w \in \mathbb{F}_q^n$ and $l \in [n]$

A q-ary code is a linear subspace C of dimension k of the vector space $\mathbb{F}_q^N$ where $\mathbb{F}_q$ is the finite field with q elements.
Informally an (r, δ, ϵ)-locally decodable code encodes k-long messages x into N-long codewords $C(x)$, such that for every $i \in [k]$, the coordinate value 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 1. *A q-ary code $C : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^N$ is said to be (r, δ, ϵ)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

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

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

where the probability is taken over the random coin tosses of $\mathcal{A}$.
2. *$\mathcal{A}$ makes at most r queries to y.*

An LDC is called *linear* if C is a linear transformation over $\mathbb{F}_q^k$. A locally decodable code allows to probabilistically decode any coordinate of a message by probing only few coordinates of its corrupted encoding. We now formally define Locally Correctable Codes (LCCs) that allows to efficiently recover not only coordinates of the message but also all the coordinates of the corrupted codeword.

Definition 2. *A code (set) C in the space $\mathbb{F}_q^N$ is said to be (r, δ, ϵ)-locally correctable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

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

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

where the probability is taken over the random coin tosses of $\mathcal{A}$.
2. *$\mathcal{A}$ makes at most r queries to y.*

Below, we prove a lemma that shows how we can build a locally decodable code if we had a locally correctable code.

Lemma 1. *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 $k = \dim C$ and set $A = \{y_1, \cdots, y_k\}$ be a basis for C. We build a $k \times N$ matrix such that every y_i is ith-row of matrix. Next, apply elementary row operations to this matrix to obtain the matrix B contains a square identity matrix as its submatrix. There are k non-zero columns in matrix B that uniquely determine an element of C. Let $I \subseteq [N]$ be a set of coordinates whose values uniquely determine an element of C. For $c \in C$ let $c|_I \in \mathbb{F}_q^k$ denote the restriction of c to coordinates in I. Given a message $x \in \mathbb{F}_q^k$ we define $C'(x)$ to be the unique element $c \in C$ such that $c|_I = x$. It is easy to see that local correctability of C yields local decodability of C'. □

3 Reed Muller codes

We start with introducing some notation. D-evaluation of a function h defined over a domain D, is a vector of values of h at all points of D. For example if $D = \{d_1, \ldots, d_n\}$ then the vector of D-evaluation of h is $\big(h(d_1), h(d_2), \ldots h(d_n)\big)$. Let $F(z_1, \ldots, z_n)$ be a polynomial of total degree at most d in the ring $\mathbb{F}_q[z_1, z_2, \ldots, z_n]$ then $\mathbb{F}_q^n$-evaluation of F is a vector with q^n coordinates that every coordinate is indexed with a vector $w \in \mathbb{F}_q^n$. The q-ary code consists of $\mathbb{F}_q^n$-evaluation of all polynomials of total degree at most d in the ring $\mathbb{F}_q[z_1, \ldots, z_n]$. These polynomials can have $\binom{n+d}{d}$ monomials and every polynomial can uniquely construct by having its coefficient. Hence, such code encodes $k = \binom{n+d}{d}$-long messages over $\mathbb{F}_q$ to q^n-long codewords.

Proposition 1. *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 $\mathbb{F}_q^n$-evaluation 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(a vector with q^n coordinates) corrupted in up to δ fraction of coordinates and a point $w \in \mathbb{F}_q^n$. The aim of the local corrector is finding $F(w)$. The local corrector picks a vector $v \in \mathbb{F}_q^n$ uniformly at random and considers a line

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

through 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 $w + \lambda v$, $\lambda \in S$ to obtain values $\{e_\lambda\}$. Next by using polynomial interpolation, it recovers the unique univariate polynomial h, $\deg h \leq d$, such that $h(\lambda) = e_\lambda$, for all $\lambda \in S$, and outputs $h(0)$.

If all $d+1$ queries of the corrector access uncorrupted locations, then h is the restriction of F to L, and $h(0) = F(w)$. The corrector will not query a corrupted coordinate with probability at least $1 - (d+1)\delta$ because at most δ fraction of coordinates of codeword (the vector with q^n coordinates corresponding to the polynomial F) are corrupted and every individual query of the corrector goes to a uniformly random location. □

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 the number of variables and the degree of polynomials and letting these parameters grow to infinity. Increasing the degree relative to the number of variables yields shorter codes of larger query complexity.

4 Multiplicity codes

Next, we study the simplest example of multiplicity codes that uses a bivariate polynomial over $\mathbb{F}_q$. In that case the codewords are indexed by bivariate polynomials of degree at most d over $\mathbb{F}_q$ and the coordinates of codewords are indexed by elements of $\mathbb{F}_q^2$.

Proposition 2. *Let q be a prime power, $\tau > 0$, and $d \leq 2(1-\tau)(q-1) - 2$ be an integer; then there exists a linear code of dimension $k = \binom{d+2}{2}$ in $\mathbb{F}_q^N$, $N = 3q^2$, that is $(2(q-1), \delta, 12\delta/\tau + 2/q)$-locally correctable for all δ.*

Proof. Codewords of the multiplicity code correspond to polynomials F in the ring $\mathbb{F}_q[z_1, z_2]$ of total degree up to d. Coordinates are organized in triples indexed by elements of $w \in \mathbb{F}_q^2$. A triple corresponding to a point w stores the values

$$F(w),\ \frac{\partial F}{\partial z_1}\Big|_w,\ \frac{\partial F}{\partial z_2}\Big|_w. \tag{1}$$

Distinct polynomials F yield distinct codewords because two distinct polynomials of degree at most d can agree on at most $\frac{d}{2q}$-fraction of the points in $\mathbb{F}_q^2$ (a fact you can see in appendix 3.6 [2]). Given a δ-corrupted codeword corresponding to a polynomial F and a point $w \in \mathbb{F}_q^2$ the local corrector needs to recover the triple (1).

- The corrector picks a vector $v_1 \in \mathbb{F}_q^2$ uniformly at random and considers a line

$$L_1 = \{w + \lambda v_1 | \lambda \in \mathbb{F}_q\}$$

through w. The goal of the corrector here is to recover the univariate restriction $f_1(\lambda) = F(w + \lambda v_1) \in \mathbb{F}_q[\lambda]$. If we had such a polynomial f_1, then we name $f_1(\lambda) = val_\lambda$ and $f_1'(\lambda) = der_\lambda$ for every $\lambda \neq 0$. The corrector queries $3(q-1)$ codeword coordinates corresponding to points $\{w + \lambda v_1\}_{\lambda \neq 0}$, to obtain the (possibly corrupted) values

$$\left\{ F(w + \lambda v_1),\ \frac{\partial F}{\partial z_1}\Big|_{w+\lambda v_1},\ \frac{\partial F}{\partial z_2}\Big|_{w+\lambda v_1} \right\}_{\lambda \neq 0}.$$

The corrector then uses these values to recover the (possibly corrupted) values $\{val_\lambda, der_\lambda\}_{\lambda \neq 0}$ of f_1 and the derivative of f_1 via the chain rule

$$f_1'(\lambda) = \frac{\partial F}{\partial z_1}\Big|_{w+\lambda v_1} v_1(1) + \frac{\partial F}{\partial z_2}\Big|_{w+\lambda v_1} v_1(2). \tag{2}$$

Next, the corrector recovers the unique univariate polynomial f_1, $\deg f_1 \leq d$, such that

$$f_1(\lambda) = val_\lambda \quad , \quad f_1'(\lambda) = der_\lambda, \tag{3}$$

for all but at most $\lfloor \tau(q-1)/2 \rfloor$ values of $\lambda \in \mathbb{F}_q^*$. This means that if for at most $\lfloor \tau(q-1)/2 \rfloor$ values of λ the equality (3) does not hold, we still can obtain a unique univariate polynomial. The uniqueness of f_1, if it exists follows from the fact that a degree d nonzero univariate polynomial cannot vanish together with its derivative at more than $d/2$ points. Suppose we obtain two distinct polynomials f_1 and f_2 that satisfy equalities (3) for at least $(q-1) - \lfloor \tau(q-1)/2 \rfloor$ values of λ. Polynomial $h = f_1 - f_2$ can not vanish together with its derivative at more than $d/2$ points , Hence we should have $(q-1) - \lfloor \tau(q-1)/2 \rfloor \leq d/2 \leq (1-\tau)(q-1) - 1$ but this inequality is not correct, so f_1 is unique. If a polynomial f_1 does not exist, the corrector halts with an arbitrary output. We make use of Markov's inequality. If X is a non-negative random variable and $a > 0$, then

$$P(X \geq a) \leq \frac{E(X)}{a}$$

where $E(X)$ is the expectation of random variable X. Let X be a random varaible that is 1 when query goes to the corrupted location and 0 otherwise. Since each individual query of the corrector accesses a uniformly random location, we can obtain $E(X) = 3\delta(q-1)$. By Markov's inequality, we have

$$P(X \geq \tfrac{\tau(q-1)}{2}) \leq \tfrac{3\delta(q-1)}{\tau(q-1)/2} = 6\delta/\tau.$$

It means the probability that $\tau(q-1)/2$ or more of the queries go to corrupted locations is at most $6\delta/\tau$. Therefore with probability at least $1 - 6\delta/\tau$, the recovered polynomial f_1 is indeed the restriction of F to the line L_1. Thus $f_1(0) = F(w)$, and $f_1'(0)$ is the derivative of F in direction v_1.

- It is not hard to see that knowing the polynomial f_1 is not sufficient to recover (1). The corrector picks a uniformly random vector $v_2 \in \mathbb{F}_q^2$ and repeats the previous step to obtain a polynomial f_2 such that $f_2'(0)$ is the derivative of F in direction v_2.
- Finally, in the last step, the corrector combines directional derivatives of F in directions v_1 and v_2 to recover the partial derivatives of F at w. It is not hard to show [3] that such a recovery is always possible whenever v_1 and v_2 are not collinear, which happens with probability at least $1 - 2/q$. Therefore with probability at least $1 - 2/q - 12\delta/\tau$ corrector recover triple (1).

□

Proposition 2 yields $O(\sqrt{k})$ -query codes of rate arbitrarily close to 2/3. General multiplicity codes are obtained by evaluating n-variate polynomials together with all their mixed partial derivatives of order up to s, for arbitrary positive integers n and s. Increasing n reduces the query complexity; increasing s yields codes of larger rate.

5 Matching Vector Codes

Matching vector codes inherit some structure from Reed Muller codes, yet there are some differences. Instead of encoding a message into $\mathbb{F}_q^n$-evaluation of all polynomials $\mathbb{F}_q[z_1, \ldots, z_n]$, they encode a message into $\mathbb{C}_m^n$-evaluation of a subset of polynomials $\mathbb{F}_q[z_1, \ldots, z_n]$ that their monomials are chosen according to a matching family of vectors, where $\mathbb{C}_m$ is a certain multiplicative subgroup of $\mathbb{F}_q^*$.

Definition 3. *Let $S \subseteq \mathbb{Z}_m \backslash \{0\}$. We say that families $U = \{u_1, \ldots, u_k\}$ and $V = \{v_1, \ldots, v_k\}$ of vectors in $\mathbb{Z}_m^n$ form an S-matching family if the following two conditions are satisfied:*

- *For all $i \in [k]$, $(u_i, v_i) = 0$;*
- *For all $i, j \in [k]$ such that $i \neq j$, $(u_i, v_i) \in S$.*

Below, we introduce some notations which will be used in this section.

Notations	
$\mathbb{C}_m$	multiplicative subgroup of $\mathbb{F}_q^*$ of order m
g	a generator of $\mathbb{C}_m$
g^w when $w \in \mathbb{Z}_m^n$	$(g^{w(1)}, \ldots, g^{w(n)}) \in \mathbb{C}_m^n$
$M_{w,v}$ when $w, v \in \mathbb{Z}_m^n$	$\mathrm{M}_{w,v} = \{g^{w+\lambda v} \mid \lambda \in \mathbb{Z}_m\}$
def height	

For a polynomial $h \in \mathbb{F}_q[z_1, \ldots, z_n]$ we denote by $\mathrm{supp}(h)$ the set of monomials with non zero coefficients in h, where a monomial $z_1^{p_1} \cdots z_n^{p_n}$ is identified

with the integer n-tuple $(p_1, \ldots, p_n)$.
For $u \in \mathbb{Z}_m^n$, we define the monomial $mon_u \in \mathbb{F}_q[z_1, \ldots, z_n]$ by

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

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

$$\begin{aligned} mon_u(g^{w+\lambda v}) &= \prod_{\ell \in [n]} \left(g^{w(\ell)+\lambda v(\ell)}\right)^{u(\ell)} \\ &= \prod_{\ell \in [n]} g^{u(\ell).w(\ell)} g^{\lambda u(\ell).v(\ell)} \\ &= g^{\sum_{\ell=1}^n u(\ell).w(\ell)} g^{\lambda \sum_{\ell=1}^n u(\ell).v(\ell)} \\ &= g^{(u,w)} (g^\lambda)^{(u,v)}. \end{aligned} \tag{5}$$

Notice that g is a generator for $\mathbb{C}_m$ and $\lambda \in \mathbb{Z}_m$, therefore the formula above implies that $M_{w,v}$-evaluation of a monomial mon_u is a $\mathbb{C}_m$-evaluation of a (univariate) monomial

$$g^{(u,w)} y^{(u,v)} \in \mathbb{F}_q[y]. \tag{6}$$

Proposition 3. *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 $m|q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long message into m^n-long codewords that is $(s+1, \delta, (s+1)\delta)$-locally decodable for all δ.*

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

$$F(z_1, \ldots, z_n) = \sum_{j=1}^{k} x(j) \cdot mon_{u_j}(z_1, \ldots z_n) \tag{7}$$

We want to prove that this code is locally decodable, so the input to the decoder is a (corrupted) $\mathbb{C}_m^n$-evaluation of F and an index $i \in [k]$ and the decoder should be able to recover $x(i)$. To achieve this, the decoder picks $w \in \mathbb{Z}_m^n$ at random, and queries the (corrupted) M_{w,v_i}-evaluation of F at $(s+1)$ consecutive locations $\{g^{w+\lambda v_i} | \lambda \in \{0, \ldots, s\}\}$ to obtain values $c_0, \ldots, c_s$. Then the decoder 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 $h(y)$ follows from standard properties of Vandermonde matrices [4].) Now we describe how the decoder can recover $x(i)$ when it has the polynomial $h(y)$. We saw that M_{w,v_i}-evaluation of a monomial mon_{u_i} is a $\mathbb{C}_m$-evaluation of a (univariate) monomial

$$g^{(u_i,w)} y^{(u_i,v_i)} \in \mathbb{F}_q[y].$$

We can replace $mon_{u_i}(z_1, \ldots, z_n)$ with this univariate monomial in equation (7) to obtain

$$f(y) = \sum_{j=1}^{k} x(j) \cdot g^{(u_j,w)} y^{(u_j,v_i)} \in \mathbb{F}_q[y]. \tag{8}$$

By knowing the properties of the S-matching family $\mathcal{U}, \mathcal{V}$ and (8), we can write

$$f(y) = x(i) \cdot g^{(u_i,w)} + \sum_{s \in S} \left(\sum_{j:(u_j,v_i)=s} x(j) \cdot g^{(u_j,w)} \right) y^s. \tag{9}$$

Hence, restriction of F to the multiplicative line M_{w,v_i} yields a univariate polynomial $f(y)$ such that the set of monomial degree of f is in $S \cup \{0\}$ and $x(i) = f(0)/g^{(u_i,w)}$.

To this end, consider the polynomial $h(y)$ that the decoder has computed. The polynomial $h(y)$ should have a structure similar to that of the polynomial $f(y)$, therefore the decoder returns $h(0)/g^{(u_i,w)}$.

If all of $(s+1)$ locations queried by the decoder are not corrupted then $h(y)$ is indeed the noiseless restriction of F to M_{w,v_i}, and decoder's outputs is correct. Because every individual query of the decoder goes to a uniformly random location, with probability at least $1 - \delta(s+1)$ the decoder returns $x(i)$. □

6 Comparison

In this section we compare three families of LDCs by focusing on their query complexity and code rate. If we apply Proposition 1 for bivariate polynomials, then we obtain a Reed Muller LDC that encodes a $\binom{2+d}{2}$-long message to q^2-long codewords. Hence, the rate of this code is

$$\frac{\binom{2+d}{2}}{q^2} \cong 1/2. \tag{10}$$

By Proposition 2, the rate of a Multiplicity Code that uses bivariate polynomials is

$$\frac{\binom{2+d}{2}}{3q^2} \cong 2/3. \tag{11}$$

Note that the rate is improved with a Multiplicity Code because of using polynomials with higher degrees(Note the relations between parameters d and q). By using polynomials with higher degree, the local decoder requires a larger number of queries. Hence, the number of queries increases in the multiplicity codes, when compared with the Reed Muller codes. Generally, Multiplicity codes improve upon Reed Muller codes in the regime of high rate but, in the regime of medium query complexity, the Reed Muller codes are better than Multiplicity codes. The parameter ϵ in the Reed Muller code is $(d+1)\delta$ and in the Multiplicity code is $12\delta/\tau + 2/q$. Let $\tau = 12/q$ and δ is equal for both codes. Next, we try to compare these codes. In that case, the code rate and number of queries are the same with above and the parameter ϵ is equal for both when q is a large number.

Matching Vector codes have shorter codeword lengths than Reed Muller codes when the query complexity (namely τ_r) is low,

$$\tau_r \leq \frac{\log k}{(\log \log k)^c},$$

for some constant c [2]. On the other hand, Matching Vector codes have longer codeword lengths than Reed Muller codes when the query complexity is high,

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

for some constant c [2].

Generally, Matching Vector codes are the best known LDCs in the regime of low query complexity, Reed Muller codes are the best known LDCs in the regime of medium query complexity, and Multiplicity codes are the best known LDCs in the regime of high query complexity [2].

7 Conclusions

In this paper we study three families of locally decodable codes. The first family, Reed Muller Codes, is constructed by using evaluation of multivariate polynomials over a finite field. The second family, Multiplicity Codes, is constructed by not only evaluation of multivariate polynomial over a finite field but also evaluation of *partial derivatives* of these polynomials. The third family, Matching Vector Codes, is constructed by using evaluation of a family of polynomials corresponding to a matching family. The basic idea of the local decoder for all of them is interpolating an univariate polynomial that agrees with a multivariate polynomial on a line.

References

[1] Yekhanin, Sergey. *Locally Decodable Codes: A Brief Survey.* http://research.microsoft.com/en-us/um/people/yekhanin/papers/survey_iwcc.pdf

[2] Yekhanin, Sergey. *Locally Decodable Codes.* http://research.microsoft.com/en-us/um/people/yekhanin/Papers/LDC_now.pdf

[3] Kopparty, Swastik and Saraf, Shubhangi and Yekhanin, Sergey. *High-rate codes with sublinear-time decoding.* http://research.microsoft.com/en-us/um/people/yekhanin/papers/highratelocal.pdf

[4] Lidl, Rudolf and Niederreiter, Harald. *Finite Fields.* Cambridge University Press, Cambridge, 1983.