A note on Efremenko’s Locally Decodable Codes

Parikshit Gopalan
MSR-Silicon Valley
parik@microsoft.com

There have been three beautiful recent results on constructing short locally decodable codes or LDCs [Yek07, Rag07, Efr09], culminating in the construction of LDCs of sub-exponential length. The initial breakthrough was due to Yekhanin who constructed 3-query LDCs of sub-exponential length, assuming the existence of infinitely many Mersenne primes [Yek07]. Raghavendra presented a clean formulation of Yekhanin’s codes in terms of group homomorphisms [Rag07]. Building on these works, Efremenko recently gave an elegant construction of 3-query LDCs which achieve sub-exponential length unconditionally [Efr09].

In this note, we observe that Efremenko’s construction can be viewed in the framework of Reed-Muller codes: the code consists of a linear subspace of polynomials in \( \mathbb{F}_q[X_1, \ldots, X_n] \), evaluated at all points in \((\mathbb{F}_q^*)^n\). We stress that this is not a new construction, but just a different view of [Efr09].

In this view, the decoding algorithm is similar to traditional local decoders for Reed-Muller codes, where the decoder essentially shoots a line in a random direction and decodes along it (see for instance [STV01]). The difference is that the monomials which are used are not of low-degree, they are chosen according to a suitable set-system. Further, the lines for decoding are multiplicative, a notion we will define shortly.

A crucial ingredient in these LDCs is a large matching set of vectors over \( \mathbb{Z}_m^n \). Such vectors can be obtained from the set-systems with restricted intersections modulo composites constructed by Grolmusz [Gro00]. His construction uses the low-degree representations of the OR function modulo composites [BBR94]. We present a construction of matching vectors directly from OR polynomials due to Sudan [Sud09], which is very simple and achieves nearly the same parameters.

1 The Code Construction.

Let \( \mathbb{F}_q \) be a finite field with \( q \) elements, \( \mathbb{F}_q^* \) its multiplicative group, and let \( m|(q - 1) \). We think of \( q \) and \( m \) as constants (say 7 and 6 for concreteness). Given \( L \subset \mathbb{Z}_m \) and an integer \( x \), we say \( x \in L \) mod \( m \) if \( x \) mod \( m \) \( \in L \).

**Definition 1.** Two families of vectors \( \mathcal{U} = \{u[1], \ldots, u[f]\} \) and \( \mathcal{V} = \{v[1], \ldots, v[f]\} \) where \( u[i], v[j] \in \mathbb{Z}_m^n \) are said to be matching if there exists \( L \subseteq \mathbb{Z}_m \setminus \{0\} \) such that
- For every \( i \in [f] \), \( u[i] \cdot v[i] = 0 \).
- For every \( i \neq i \in [f] \), \( u[i] \cdot v[j] \in L \) mod \( m \).

If \( m \) is a prime power, then \( f \) can be at most polynomial in \( n \) [Gop06]. For composite \( m \) with two or more prime factors, Grolmusz shows that \( f \) can be super-polynomial in \( n \) [Gro00].
Lemma 2. If $m$ has $t$ distinct prime factors, then there is an (explicit) matching family $U, V$ of subsets of vectors in $\mathbb{Z}_m^n$ such that $\ell = |L| \leq 2^t - 1$ and $f \geq \exp\left(\frac{(\log n)^f}{(\log \log n)^{t-1}}\right)$.

We now describe the code $C = C(U, V)$.

- **Message Space:** For each vector $v[j] \in V$, define the monomial $\chi_j(x) = \prod_{k \in [n]} x_k^{v[j]k}$. Messages correspond to polynomials $P(x) = \sum_{j \leq f} \lambda_j \chi_j(x)$ where $\lambda_i \in \mathbb{F}_q$.

- **Encoding:** The encoding is the evaluation of the polynomial $P$ at all points in $(\mathbb{F}_q^n)$.

It follows that $C_x$ is linear over $\mathbb{F}_q$, it has dimension $f$ and length $(q - 1)^n$. We will give a local decoder for it with query complexity $\ell + 1$.

**The Local Decoder.** Let $\gamma$ be an element of order $m$ in $\mathbb{F}_q^*$. We define the set $B = \{c | c \in L\} \subset \mathbb{F}_q^*$. Note that $1 \notin B$. For a scalar $\lambda \in \mathbb{F}_q^*$ and a vector $u \in \mathbb{Z}_m^n$, let $\lambda^u = (\lambda^{u_1}, \ldots, \lambda^{u_n})$ and more generally $\lambda^{hu} = (\lambda^{hu_1}, \ldots, \lambda^{hu_n})$ for $h \in \mathbb{Z}$. For two vectors $x, y \in (\mathbb{F}_q^n)$ we use $x \odot y$ to denote the vector $(x_1y_1, x_2y_2, \ldots, x_ny_n) \in (\mathbb{F}_q^n)$.

With this notation set up, let us define the multiplicative line through $x \in (\mathbb{F}_q^n)$ in the direction $\gamma^u$ as the set of points $\{x, \gamma^u \odot x, \gamma^{2u} \odot x, \ldots\} \in (\mathbb{F}_q^n)$. The following Lemma which shows that $\chi_i$ is the unique monomial that stays constant along this line, is the key to decoding.

**Lemma 3.** For any $i, j \in [f], h \in \mathbb{Z}$ and $x \in (\mathbb{F}_q^n)$,

$$\chi_j(\gamma^hu[i] \odot x) = \begin{cases} \chi_j(x) & \text{if } i = j \\ \beta^h\chi_j(x) & \text{for } \beta \in B \text{ if } i \neq j. \end{cases}$$

**Proof.** We will prove the claim when $h = 1$, the general case is similar. We have

$$\chi_j(\gamma^u[i] \odot x) = \prod_{k \in [n]} (\gamma^u[i]k x_k)^{v[j]k} = \gamma^{\sum_k u[i]k v[j]k} \prod_{k \in [n]} x_k^{v[j]k} = \gamma^{u[i] \odot v[j]} \chi_j(x).$$

If $i = j$, then $u[i] \cdot v[j] \equiv 0 \mod m$, and $\gamma^{u[i] \odot v[j]} = 1$. Whereas if $i \neq j$, then $u[i] \cdot v[j] \in L$, hence $\gamma^{u[i] \odot v[j]} = \beta \in B$, which completes the proof.

We need the following claim from [Efr09]:

**Claim 4.** There exist $c_0, \ldots, c_\ell \in \mathbb{F}_q$ such that $\sum_{h=0}^{\ell} c_h = 1$ and $\sum_{h=0}^{\ell} c_h \mu^h = 0$ for $\mu \in B$.

The $c_h$s are the coefficients of a univariate polynomial that vanishes on $B$, suitably rescaled.

We now state the decoding algorithm. The algorithm has query access to $P$ and is given $i \in [f]$ as input. The goal is to return $\lambda_i$.

1. Pick $x \in (\mathbb{F}_q^n)$ at random, query the values $P(x), P(\gamma^u[i] \odot x), \ldots, P(\gamma^{\ell u}[i] \odot x)$.

2. Return $(\sum_{h=0}^{\ell} c_h P(\gamma^hu[i] \odot x)) \cdot (\chi_i(x))^{-1}$.

In step 2, the algorithm needs to compute $\chi_i(x)^{-1}$, which is easy given $i$ and $x$.

**Theorem 5.** The Decoding Algorithm returns the coefficient $\lambda_i$. 

2
Definition 6. A polynomial directly uses representations of the OR function. We present a simple construction of matching vectors due to Madhu Sudan. The construction

Equation 1 depends on the index $j$, but we suppress this for notational clarity.

Proof. We have

$$
\sum_{h=0}^{\ell} c_h P(\gamma^h u[i] \circ x) = \sum_{h=0}^{\ell} c_h \sum_{j \in [\ell]} \lambda_j x_j (\gamma^h u[i] \circ x) = \sum_{j \in [\ell]} \lambda_j \sum_{h=0}^{\ell} c_h x_j (\gamma^h u[i] \circ x)
$$

$$
= \sum_{j \neq i \in [\ell]} \lambda_j \sum_{h=0}^{\ell} c_h \beta^k x_j (x) + \lambda_i \sum_{h=0}^{\ell} c_h x_j (x)
$$

$$
= \lambda_i x_i (x)
$$

where Equation 1 uses Lemma 3, and Equation 2 uses Claim 4. We note that $\beta = \gamma \circ [i]:\circ [j]$ in Equation 1 depends on the index $j$, but we suppress this for notational clarity.

With Grolmusz’s construction, the code $C_X$ gives encoding length $(q-1)^n$, dimension $f = n^{\omega(1)}$ and query complexity $2^t$. Put differently, messages of length $k$ are encoded by codewords of length $\exp(\exp(O((\log k)^{\tilde{\ell}} (\log \log k)^{1-\frac{1}{t}})))$, which can be decoded using $2^t$ queries.

2 Sudan’s construction of matching vectors.

We present a simple construction of matching vectors due to Madhu Sudan. The construction directly uses representations of the OR function.

Definition 6. A polynomial $P(X_1, \ldots, X_k)$ represents the OR function on $\{0,1\}^k$ modulo $m$ if there exists $L \subseteq \mathbb{Z}_m \setminus \{0\}$ so that $P(0^k) \equiv 0 \mod m$ and $P(x) \equiv L \mod m$ for every $x \in \{0,1\}^k \setminus \{0^k\}$.

Barrington et al. [BBR94] proved a surprising upper bound on the degree of such polynomials.

Theorem 7. [BBR94] If $m$ has $t$ distinct prime divisors, there is a polynomial of degree $O(k^{1\over t})$ representing the OR function on $\{0,1\}^k$. The constant hidden by the $O$ term depends only on $m$.

We give a construction of a large matching family of vectors from any low-degree OR polynomial $P(X_1, \ldots, X_k)$. The vectors are indexed by $y \in \{0,1\}^k$. Given such a $y$, define the polynomial $P_y(X_1, \ldots, X_k)$ by replacing $X_i$ in $P$ with $1 - X_i$ if $y_i = 1$, and leaving $X_i$ unchanged if $y_i = 0$. Thus $P_y$ is just $P$ with the origin shifted to $y$. Given vectors, $x, y \in \{0,1\}^n$, we use $x \oplus y$ to denote the bitwise Xor of the two vectors. The following properties of $P_y$ are easy to verify

Lemma 8. $P_y$ is multilinear, with $\deg(P_y) = \deg(P)$. For $x \in \{0,1\}^k$, $P_y(x) = P(x \oplus y)$.

Let $\deg(P) = d$. Set

$$
n = \sum_{i \leq d} \binom{k}{i}.
$$

We construct matching families of size $f = 2^k$ in $n$ dimensions as follows:

- The family $\mathcal{U}$ is indexed by vectors $x \in \{0,1\}^k$. For each such $x$, the vector $u[x]$ is obtained by evaluating all multilinear monomials of degree at most $d$ at $x$, so $u[x]$ has dimension $n$.

- The family $\mathcal{V}$ is indexed by vectors $y \in \{0,1\}^k$. For each such $y$, $P_y$ is a multilinear polynomial of degree $d$. The vector $v[y]$ is the vector of its coefficients (which also has dimension $n$).
Theorem 9. The families $\mathcal{U}$ and $\mathcal{V}$ are a matching family of vectors.

Proof. The key observation is that $u[x] \cdot v[y] = P_y(x)$, since $v[y]$ gives the coefficients of each monomial, and $u[x]$ gives the evaluations of these monomials at $x$. By Lemma 8, $P_y(x) = P(x \oplus y)$. By Definition 6, $P_y(y) = P(0^k) \equiv 0 \mod m$, whereas $P_y(x) = P(x \oplus y) \in L \mod m$ for all $x \neq y \in \{0, 1\}^k$.

The BBR construction gives $d = O(\sqrt{k})$. Plugging this in yields $n = k^{O(\sqrt{k})} = 2^{O(\sqrt{k} \log k)}$.

Summary. A better construction of matching vectors will give LDCs with better parameters. Known constructions of matching vectors rely on the low-degree polynomials representing the OR function modulo composites, discovered by Barrington et al. [BBR94]. These polynomials have now found diverse combinatorial applications; LDCs, set-systems and Ramsey graphs to name a few, yet there is an exponential gap in the known degree bounds for these polynomials [Gop06]. There is also no strong evidence for what the right bound should be. We pose closing this gap as a natural open question.

Acknowledgments. I thank Madhu Sudan for allowing me to include his construction of matching vectors in this writeup. I thank Venkatesan Guruswami, Prasad Raghavendra, Sergey Yekhanin and Klim Efremenko for useful discussions, and Sergey again for encouraging me to write this note.

References


