Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 69 (2009)

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 $\mathbb{F}_q[X_1, \dots, X_n]$, evaluated at all points in $(\mathbb{F}_q^\star)^n$. 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, $\mathbb{F}_q^\star$ 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 $x \in L \bmod m$ if $x \bmod m \in L$.

Definition 1. Two families of vectors $\mathcal{U} = \{u[1], \dots, u[f]\}$ and $\mathcal{V} = \{v[1], \dots, v[f]\}$ 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 $i \neq i \in [f]$, $u[i] \cdot v[j] \in L \bmod m$.

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].

ISSN 1433-8092

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 $\ell = |L| \le 2^t - 1$ and $f \ge \exp\left(\frac{(\log n)^t}{(\log\log n)^{t-1}}\right)$.*

We now describe the code $\mathcal{C} = \mathcal{C}(\mathcal{U}, \mathcal{V})$.

- **Message Space:** For each vector $v[j] \in \mathcal{V}$, define the monomial $\chi_j(x) = \prod_{k\in[n]} x_k^{v[j]_k}$. Messages correspond to polynomials $P(x) = \sum_{j\le f} \lambda_j \chi_j(x)$ where $\lambda_i \in \mathbb{F}_q$.
- **Encoding:** The encoding is the evaluation of the polynomial P at all points in $(\mathbb{F}_q^\star)^n$.

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

With this notation set up, let us define the *multiplicative* line through $x \in (\mathbb{F}_q^\star)^n$ in the direction $\gamma^{u[i]}$ as the set of points $\{x, \gamma^{u[i]} \odot x, \gamma^{2u[i]} \odot x, \dots\} \in (\mathbb{F}_q^\star)^n$. The following Lemma which shows that χ_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 $x \in (\mathbb{F}_q^\star)^n$,*

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

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

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

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

We need the following claim from [Efr09]:

Claim 4. *There exist $c_0, \dots, c_\ell \in \mathbb{F}_q$ 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_hs 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 λ_i.

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

Proof. We have

$$\begin{aligned}\sum_{h=0}^{\ell} c_h P(\gamma^{hu[i]} \odot x) &= \sum_{h=0}^{\ell} c_h \sum_{j\in[f]} \lambda_j \chi_j(\gamma^{hu[i]} \odot x) \;\; = \sum_{j\in[f]} \lambda_j \sum_{h=0}^{\ell} c_h \chi_j(\gamma^{hu[i]} \odot x) \\ &= \sum_{j\neq i\in[f]} \lambda_j \sum_{h=0}^{\ell} c_h \beta^h \chi_j(x) + \lambda_i \sum_{h=0}^{\ell} c_h \chi_j(x) \qquad (1) \\ &= \lambda_i \chi_i(x) \qquad (2)\end{aligned}$$

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

With Grolmusz's construction, the code $\mathcal{C}_{\mathcal{F}}$ 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 $\exp(\exp(O((\log k)^{\frac{1}{t}}(\log\log k)^{1-\frac{1}{t}})))$, 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 $P(0^k) \equiv 0 \bmod m$ and $P(x) \in L \bmod m$ 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 $O(k^{\frac{1}{t}})$ 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 cofficients 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, $P_y(y) = P(0^k) \equiv 0 \bmod m$, whereas $P_y(x) = P(x \oplus y) \in L \bmod m$ for all $x \neq y \in \{0,1\}^k$. $\square$

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

[BBR94] David A. Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994. 1, 3, 4

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pages 39–44, 2009. 1, 2

[Gop06] Parikshit Gopalan. *Computing with Polynomials over Composites*. PhD thesis, Georgia Institute of Technology, 2006. 1, 4

[Gro00] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20(1):71–86, 2000. 1

[Rag07] Prasad Raghavendra. A note on Yekhanin's locally decodable codes. *Electronic Colloqium on Computational Complexity (ECCC)*, TR07-016, 2007. 1

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001. 1

[Sud09] Madhu Sudan. Personal Communication, 2009. 1

[Yek07] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of ACM*, pages 1–16, 2007. 1

ECCC ISSN 1433-8092

http://eccc.hpi-web.de