1 Definition of Locally Decodable/Correctable Codes

In traditional coding theory, one encodes a message $m \in \{0,1\}^k$ to a codeword $c \in \{0,1\}^n$. The codeword then gets corrupted on δ fraction of symbols. Our goal is to recover m or c using the corrupted codeword $\tilde{c}$. In this setting, for example, we have expander code with $O(n)$ decoding time (and even $O(n)$ encoding time [SS96]).

One natural question is: what if we only want to recover just one bit of m or c? Can this be done very efficiently? Hopefully, we want to achieve this by only looking at a very small number of symbols (e.g., $O(1)$ symbols) in $\tilde{c}$?

Note that we can not hope to do this deterministically, since the δ fraction of errors can happen anywhere, including the $O(1)$ places we care about. Thus, we may want to do this randomly.

Here follows formal definitions of locally decodable codes and locally correctable codes.

Definition 18.1 (Locally Decodable Codes)**.** A code $C : \{0,1\}^k \to \{0,1\}^n$ is said to be a (q, δ, ε)-Locally Decodable Code if and only if $\forall i \in [k]$, there exists a randomized algorithm $\mathcal{A}_i$ such that $\forall m \in \{0,1\}^k, \forall \tilde{c} \in \{0,1\}^n$,

- $\mathcal{A}_i$ randomly queries $\leq q$ coordinates of $\tilde{c}$,
- if $\Delta(\tilde{c}, C(m)) \leq \delta n$, then $\mathbf{Pr}[\mathcal{A}_i(\tilde{c}) = m_i] \geq \frac{1}{2} + \varepsilon$.

Here δ, ε are usually constants, and q is called query complexity.

The definition of Locally Correctable Codes is almost the same as Locally Decodable Codes, except m_i (the bit we want to recover) is replaced with $C(m)_i$.

Definition 18.2 (Locally Correctable Codes)**.** A code $C : \{0,1\}^k \to \{0,1\}^n$ is said to be a (q, δ, ε)-Locally Correctable Code if and only if $\forall i \in [k]$, there exists a randomized algorithm $\mathcal{A}_i$ such that $\forall m \in \{0,1\}^k, \forall \tilde{c} \in \{0,1\}^n$,

- $\mathcal{A}_i$ randomly queries $\leq q$ coordinates of $\tilde{c}$,
- if $\Delta(\tilde{c}, C(m)) \leq \delta n$, then $\mathbf{Pr}[\mathcal{A}_i(\tilde{c}) = C(m)_i] \geq \frac{1}{2} + \varepsilon$.

There are two remarks.

Remark 18.3. A linear (q, δ, ε)-LCC is also a linear (q, δ, ε)-LDC, because linear codes can be made systematic. For general codes, [BGT17] proves that a LCC is also a LDC, with an $O(1)$ factor off in query complexity q. This means Locally Correctable Code is in some sense a stronger notion than Locally Decodable Code.

Remark 18.4. We can assume $\mathcal{A}_i$'s are non-adaptive by changing ε to $\varepsilon/2^q$. Non-adaptive means the j-th query does not depend on previous queries. The idea is to guess the q answers in advance, then run $\mathcal{A}_i$ based on those answers. If the queried q bits match the guessed q bits, which happens with probability $\frac{1}{2^q}$, then we are happy and we get an ε advantage compared to randomly guessing. Otherwise we just output a random bit as the answer. Note that the binary alphabet is important here. When alphabet is larger, we need to be more careful.

2 Hadamard Code as Locally Correctable Code

Recall that Hadamard Code is a mapping from $\{0,1\}^k$ to $\{0,1\}^n$, where $n = 2^k$, and m maps to $\mathrm{Had}(m) = (\langle a, m\rangle)_{a\in\mathbb{F}_2^k}$.

Claim 18.5. *Hadamard Code is a $(2, \delta, 1-2\delta)$-Locally Correctable Code.*

Proof. For any $a \in \{0,1\}^k$, we wan to recover $\langle a, m\rangle$. Given $\tilde{c} \in \{0,1\}^n$ such that $dist(\tilde{c}, \mathrm{Had}(m)) \leq \delta n$, the algorithm $\mathcal{A}_a$ runs as follows:

1. Pick $z \in \{0,1\}^k$ uniformly at random.
2. Query $\tilde{c}$ at z and $a+z$.
3. Return $\tilde{c}_z + \tilde{c}_{a+z}$.

Let $c = \mathrm{Had}(m)$ be the correct codeword, then we clearly have

$$c_z + c_{a+z} = \langle z, m\rangle + \langle a+z, m\rangle = \langle a, m\rangle.$$

Although correlated, $z, a+z$ are individually uniform over $\{0,1\}^k$. Given that there are only δ fraction of errors in $\tilde{c}$, we have

$$\mathbf{Pr}[\tilde{c}_z \neq c_z] \leq \delta \wedge \mathbf{Pr}[\tilde{c}_{a+z} \neq c_{a+z}] \leq \delta.$$

By union bound,

$$\begin{aligned}
&\mathbf{Pr}[\mathcal{A}_a \text{ is correct}] \\
=&\mathbf{Pr}[\tilde{c}_z + \tilde{c}_{a+z} = \langle a, m\rangle] \\
\geq&\mathbf{Pr}[\tilde{c}_z = c_z \wedge \tilde{c}_{a+z} = c_{a+z}] \\
\geq&1 - 2\delta
\end{aligned}$$

□

3 Reed-Muller Code as Locally Correctable Code

Recall that Reed-Solomon Code maps a univariate polynomial $f \in \mathbb{F}[x]_{\leq k}$ to its evaluation on a set of points: $(f(a_1), \ldots, f(a_n))$. Reed-Muller Code just changes the univariate polynomial to a multivariate polynomial.

Definition 18.6. Let $q > d$, Reed-Muller Code $\mathrm{RM}[q, m, d]$ maps an m-variable degree d polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_m]_{\leq d}$ to its evaluation on all points. Specifically,

$$\mathrm{RM}[q, m, d] = \{(f(a_1, \ldots, a_m))_{a_1,\ldots,a_m\in\mathbb{F}_q} | f \in \mathbb{F}_q[x_1, \ldots, x_m]_{\leq d}\}$$

It's easy to compute Reed-Muller Code's parameters:

- $n = q^m$.
- $k = \binom{m+d}{m}$ (which is the number of solutions of m non-negative numbers summing up $\leq d$).
- $D \geq n \cdot \left(1 - \frac{d}{q}\right)$ (from Schwartz-Zippel Lemma).

Note that the parameters are inferior to that of the Reed-Solomon Code. But the point is that Reed-Solomon codeword is indexed by $\mathbb{F}_q$, while Reed-Muller codeword is indexed by $\mathbb{F}_q^m$, which has a much richer geometry. This endows Reed-Muller Code with good locally correctability. Furthermore, for constant query complexity, Reed-Muller Code is the state-of-the-art LCC.

Claim 18.7. *$RM[q, m, d]$ is a $[d+1, \delta, 1-(d+1)\delta]$-Locally Correctable Code.*

Proof. Given some $\vec{a} \in \mathbb{F}_q^m$, we want to recover $f(\vec{a})$ based on a corrupted codeword $\tilde{c} \in \mathbb{F}_q^n$ such that $||\tilde{c} - \mathrm{RM}(f)||_0 \leq \delta n$. The intuition is that f restricted to any line is a univariate polynomial of degree at most d. Then we can use $d+1$ values on a line which goes through $\vec{a}$, to interpolate $f(\vec{a})$. The algorithm $\mathcal{A}_{\vec{a}}$ goes as follows.

1. Randomly pick $\vec{b} \in \mathbb{F}_q^m \setminus \{0^m\}$, which is the direction of the line.
2. Randomly pick distinct $\lambda_1, \ldots, \lambda_{d+1} \in \mathbb{F}_q$.
3. Query $\tilde{c}$ at $\vec{a} + \lambda_1 \vec{b}, \ldots, \vec{a} + \lambda_n \vec{b}$.
4. Interpolate a polynomial $A \in \mathbb{F}_q[x]_{\leq d}$ such that $A(\lambda_i) = \tilde{c}_{\vec{a}+\lambda_i \vec{b}}, \forall i \in [d+1]$.
5. Return $A(0)$.

Like before, each $\vec{a} + \lambda_i b$ is uniform over $\mathbb{F}_q^m$ although they are correlated. Thus we have

$$\forall i \in [d+1], \mathbf{Pr}[\tilde{c}_{\vec{a}+\lambda_i \vec{b}} \neq f(\vec{a} + \lambda_i \vec{b})] \leq \delta.$$

By union bound,

$$\mathbf{Pr}[\forall i \in [d+1], \tilde{c}_{\vec{a}+\lambda_i \vec{b}} = f(\vec{a} + \lambda_i \vec{b})] \geq 1 - (d+1)\delta.$$

Define $g(x) = f(\vec{a} + x\vec{b})$, the facts $\deg(g) \leq \deg(f) = d$ and $g = A$ on $d+1$ points imply that $g = A$. Therefore, $A(0) = g(0) = f(\vec{a})$ with probability at least $1 - (d+1)\delta$. □

Here is a list which summarizes the trade-off between q and n of Reed-Muller Codes.

q	n
$O(1)$	$\exp(O(k^{\frac{1}{q-1}}))$
$\log n$	$k^{O(\log \log k)}$
$\log^t n, t > 1$	$k^{1+\frac{1}{t-1}+o(1)}$
$n^{1/t}, t \geq 1$	$t^{t+o(t)} \cdot k$

4 A Lower Bound on Linear LDC

Think of $q = O(1)$ and a linear Locally Decodable Code C. [KT00] proves that $n \geq \Omega(k^{\frac{1}{q-1}})$. The idea is to pick a random subset $S \subseteq [n]$ of size $n^{1-\frac{1}{q}}$, and show that $C(m)|_S$ almost determines m with high probability. Then $|S| \geq k \Rightarrow n^{1-\frac{1}{q}} \geq k \Rightarrow n \geq k^{\frac{q}{q-1}}$.

Here is a useful characterization of linear LDCs:

Claim 18.8. *A linear code $C : \mathbb{F}_2^k \to \mathbb{F}_2^n$ is a (q, δ)-LDC if $\forall i \in [k]$, there exists a q-uniform matching of $[n]$, $\mathcal{M}_i = \{S_1, \ldots, S_{\delta n}\}$ (i.e., S_j's are disjoint subsets of $[n]$ and $|S_j| = q$), such that*

$$\sum_{t \in S_j} G_t = e_i, \forall j \in [\delta n].$$

Here G_t is the t-th row of the generator matrix of C, e_i is the i-th unit vector in $\mathbb{F}_2^k$.

Claim 18.9. *Pick $S \subseteq [n]$ in the way that include every $s \in [n]$ into S independently with probability $n^{-\frac{1}{q}}$. Then*

$$\mathbf{Pr}[\exists j \in [\delta n] : S_j \subseteq S]] \geq 1 - O\left(\frac{1}{n}\right).$$

5 Notes

Locally Decodable Codes existed since 90's, but was formally defined after 2000. [Yek12] is a nice survey on LDC.

There are some newer constructions compared to the Reed-Muller Codes, like

- Lifted Code [GKS13].
- Multiplicity Codes [KSY14].
- Expander Codes [HOW15].

All the above can achieve $q = O(n^{\varepsilon})$ and $R \geq 1 - \varepsilon$, while for Reed-Muller Codes, $q \geq \sqrt{n}$ and $R \leq \frac{1}{2}$.

[KMRS17] has a construction for LDC with large alphabet. They reach $q = 2^{O(\sqrt{\log n})}, R \geq 1 - \varepsilon$, but the alphabet they use is very large.

References

[BGT17] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 30:1–30:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 18.3

[GKS13] Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 529–540. ACM, 2013. 5

[HOW15] Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Information and Computation*, 243:178–190, 2015. 40th International Colloquium on Automata, Languages and Programming (ICALP 2013). 5

[KMRS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *J. ACM*, 64(2):11:1–11:42, 2017. 5

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28:1–28:20, 2014. 5

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 80–86. ACM, 2000. 4

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inf. Theory*, 42(6):1710–1722, 1996. 1

[Yek12] Sergey Yekhanin. Locally decodable codes. *Found. Trends Theor. Comput. Sci.*, 6(3):139–255, 2012. 5