Coding Theory

Lecture 10

Lecture by Dr. Elette Boyle
Typeset by Steven Karas

2018-01-02
Last edited 18:22:10 2018-01-02

Disclaimer These lecture notes are based on the lecture for the course Coding Theory, taught by Dr. Elette Boyle at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture notes written by Dr. Elette Boyle.

Agenda

- Locally Decodable Codes (self-synchronizing)
- Hadamard Code
- Reed-Muller Codes
- Survey of known codes
- Cryptographic applications

1 Review

Last time we finished covering decoding binary expander codes. Expander codes are a subset of Low Density Parity Check (LDPC).

Note that we can also consider more generic constructions of codes from bipartite graphs. Each parity vertex $r \in R$ gives a more general constraint on its neighbors. Given a total ordering on the vertices, we can define a simple single-parity check code by defining the neighbors of r as a

2 Locally Decodable Codes (LDC)

Intuition Suppose that we start with a very long data stream m. Instead of recovering the entire data stream, we simply want to robustly recover some subset of symbols.

A simple way to do this is to decode the entire codeword, look up the symbols we want, and then throw away the rest. But we want to recover these symbols in sublinear time.

We could encode the data stream into blocks, which is good, but very poor error correction guarantees.

Locally Correctable Codes (LCC) are similar, but the goal is to recover symbols from the codeword itself.

2.1 Formally

A q-ary code $C : [q]^k \to [q]^n$ is a (r, δ, ε) locally decodable code if there exists a randomized decoding algorithm $\text{Dec} : [q]^n \times [k] \to [q]$ that is robust and local.

Robustness is defined as if for any index $i \in [k]$ we are able to recover from some received message $\vec{y} \in [q]^n$ within $\Delta(C(\vec{x}), \vec{y}) \leq \delta n$ distance of the original message $\vec{x} \in [q]^k$ the symbol x_i with probability $\geq 1 - \varepsilon$:

$$\Pr[\text{Dec}(\vec{y}, i) = x_i] \leq 1 - \varepsilon$$

Locality is defined as reading at most r symbols of $\vec{y}$.

A note on the parameters Generally speaking, we want r to be small, δ to be big, and ε to be small.

For data transmission, we consider δ to be constant (e.g. $\sim 1/4$), but we want n to be small, and we don't care so much about r, so long as $r \leq k$.

For cryptographic applications, we don't care much about δ, so long as $\delta > 0$ and $\varepsilon < 1/2$ is held as constant. Mostly cares about the tradeoff between r and n.

2.2 Hadamard Code

Recall that $C_{\text{Had}} : \{0,1\}^k \to \{0,1\}^{2^k}$.

$$\vec{m} \to (\vec{m}, \vec{v})_{\vec{v} \in \{0,1\}^k}$$

Indexed by $\vec{y} \in \{0,1\}^{2^k}$ by vectors $\vec{v} \in \{0,1\}^k$.

We can recover m_i by considering both $\vec{v}$ and $\vec{v} \oplus \vec{e_i}$ where $\vec{e_i}$ is the unit vector with position i:

$$\vec{m} \cdot (\vec{v} \oplus \vec{e_i}) = \vec{m} \cdot \vec{v} \oplus \vec{m} \cdot \vec{e_i} = (\vec{m} \cdot \vec{v}) \oplus m_i$$

$$m_i = \vec{m} \cdot \vec{v} \oplus \vec{m} \cdot (\vec{v} \oplus \vec{e_i})$$

Proof The $[\overbrace{2^k}^{n}, \overbrace{k}^{k}]$ Hadamard Code is $(r = 2, \delta, 2\delta)$ locally decodable.

Consider the following randomized decoding algorithm:

Local decoding Hadamard

```
Dec($\vec{y}$, $i$):
    Sample a random vector $\vec{v} \leftarrow \{0,1\}^k$
    $y_{\vec{v}}$ = value of $\vec{y}$ at position $\vec{v}$
    $y_{\vec{v}\oplus\vec{e_i}}$ = value of $\vec{y}$ at position $\vec{v} \oplus \vec{e_i}$
    return $(y_{\vec{y}}) \oplus (y_{\vec{v}\oplus\vec{e_i}}) \in \{0,1\}$
```

Note that if the $\vec{v}$ and $\vec{v} \oplus \vec{e_i}$ positions of $\vec{y}$ do not have error, then:

$$\begin{aligned}(y_{\vec{v}}) \oplus (y_{\vec{v}\oplus\vec{e_i}}) &= (\vec{m} \cdot \vec{v}) \oplus (\vec{m} \cdot (\vec{v} \oplus \vec{e_i})) \\ &= \vec{m} \cdot \vec{v} \oplus \vec{m} \cdot \vec{v} \oplus \vec{m} \cdot \vec{e_i} \\ &= m_i\end{aligned}$$

So $\Pr_{\vec{v}}[\text{Dec}(\vec{y}, i) = m_i] \geq \Pr_{\vec{v}}[\vec{v}, \vec{v} \oplus \vec{e_i} \text{ not in error}]$.

Let $S \subseteq \{0,1\}^n$ be the error positions in $\vec{y}$. Suppose that $|S| \leq \delta n$.

$$\Pr_{\vec{v}}[\vec{v} \in S] = \frac{|S|}{n} \leq \delta$$

$$\Pr_{\vec{v}}[\vec{v} \oplus \vec{e_i} \in S] = \frac{|S|}{n} \leq \delta$$

$$\Pr_{\vec{v}}[(\vec{v} \in S) \text{ or } (\vec{v} \oplus \vec{e_i} \in S)] \leq 2\delta$$

Therefore, $\Pr_{\vec{v}}[\text{Dec is correct}] \geq 1 - 2\delta$

From the union bound, it follows that the aggregate error is less than 2δ.

2.3 Systematic Codes

As a side note, k field elements can be construed as a deg $\leq k-1$ polynomial. Either via coefficients or via the 1st k evaluations of the polynomial (and interpolating the rest of the codeword). This has the advantage of including the original message as the first k elements of the codeword. Codes where codewords include the original message are called *systematic codes*.

For a systematic code, local correction implies local decoding.

Formally A code $C : [q]^k \to [q]^n$ is systematic if for every subset $S \subseteq [n]$ of size k such that $\forall \vec{x} \in [q]^k$, the codeword $C(\vec{x})$ restricted to coordinates in S is equal to $\vec{x}$.

2.4 Reed Solomon as an LDC

Take some Reed Solomon code with large k. This defines a deg $\leq k-1$ polynomial:

$$f_{\vec{m}} = \sum_{i=0}^{k-1} m_i x^i$$

$$\vec{m} \to (f_{\vec{m}}(\alpha_1), \ldots, f_{\vec{m}}(\alpha_n))$$

For any $k-1$ symobls of the codeword (excluding $\alpha = i$) gives us no information on m_i. Any value of $m_i \in \mathbb{F}_q$ is equally likely.

As a result, Reed Solomon is not a good LDC.

2.5 Reed-Muller Codes

Reed-Muller codes are a generalization of Reed-Solomon codes to higher dimensions (multi-variate polynomials).

Effectively, each codeword is a surface in this higher dimension.

Formally A Reed-Muller code has 3 parameters: q - field size, m dimensions (number of variables), and ℓ degree bound. The corresponding RM code consists of all evaluations (inputs ranging over ${\mathbb{F}_q}^m$) of all polynomials of total degree bounded by ℓ in $\mathbb{F}_q[x_1, \ldots, x_m]$.

This means that codewords are low degree polynomials.

The block length $n = q^m$ (all ${\mathbb{F}_q}^m$ input evals).

The dimension of this code is $k = \binom{m+\ell}{\ell}$ which is the number of free coefficients in such a polynomial. [1]

The distance of the code is $d = \left(1 - \frac{\ell}{q}\right) q^m$

Schwartz Lemma Let $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ be a non-zero polynomial of total degree $\leq \ell < q$.

$$\Pr_{(a_1,\ldots,a_m) \leftarrow {\mathbb{F}_q}^m}[f(a_1, \ldots, a_m) = 0] \leq \frac{\ell}{q}$$

The full proof is by induction on the number of variables.

This is another way of stating the bound on the number of roots.

2.5.1 Local Correction

Let $m, \ell \in \mathbb{N}$ and q is a prime power. We claim there exists a linear code of dimension $k = \binom{m+\ell}{\ell}$ in ${\mathbb{F}_q}^n$ where $n = q^m$ that is a $(\ell+1, \delta, (\ell+1)\delta)$ locally correctable code for any $\delta > 0$.

Given oracle access to $\vec{y} =$ evals on all of ${\mathbb{F}_q}^m$ corrupted from a codeword in at most δ fraction of coordinates. Also given a point $\vec{w} \in {\mathbb{F}_q}^m$ which corresponds to the index of the codeword we wish to recover.

Choose a random line through $\vec{w}$ in ${\mathbb{F}_q}^m$ (the floor) for some randomly chosen vector $\vec{v} \in {\mathbb{F}_q}^m$

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

Let S be an arbitrary subset of $\mathbb{F}_q \setminus \{0\}$ with $|S| = \ell + 1$. This is the number of evaluation points in the line needed to interpolate the polynomial.

Query $\vec{y}$ at the $\ell+1$ coordinates corresponding to these points on the line:

$$\{\vec{w} + \lambda\vec{v} \mid \lambda \in S\} \subseteq L$$

Denote the answers by $\{e_\lambda\}_{\lambda \in S}$ where $e_\lambda \in \mathbb{F}_q$ is the evaluation at $\vec{w} + \lambda\vec{v}$.

Note that the induced cross section (plane as defined by L) gives us a Reed-Solomon code.

Recover the unique univariate polynomial $h \in \mathbb{F}_q(x)$ with $\deg h \leq \ell$ such that $h(\lambda) = e_\lambda \; \forall \lambda \in S$.

Output $h(0)$.

[1] Can be thought of as distributing ℓ balls into $m+1$ bins representing $x_m, \ldots, x, 1$ monomials

Proof of ε If there are no errors in the queried values $\{e_\lambda\}_{\lambda \in S}$, then we get a correct recovery.

$$\Pr_{\vec{v}}[\text{ there is an error in the queries}] \leq \sum_{\lambda \in S} \Pr_{\vec{v}}[\text{there is an error not in the eval point}]$$

Each $\vec{w} + \lambda\vec{v}$ is uniform over ${\mathbb{F}_q}^m$.
Therefore, we have less than $\leq |S| \frac{\#\text{ error locations}}{\#\text{ total lcoations}}$.
This implies that $\leq (\ell + 1)\delta$ probability of accurate correction.

3 Next Time

Applications to cryptography