CSS.413.1 Topics in Coding Theory

Instructor: Mrinal Kumar

TIFR 2025, Aug-Nov

Scribe: Soham Chatterjee

soham.chatterjee@tifr.res.in
Website: sohamch08.github.io

Contents

1 Targets

The content of this course will be the followings:

- Introduction to Coding Theory: Definitions, Basic Properties, Linear Codes
- Reed Solomon Codes, Reed Muller Codes
- Decoding algorithms for Reed Solomon Codes:
  - Barlekamp-Welch Algorithm
  - Sudan's List Decoding Algorithm
  - Guruswami-Sudan List Decoding Algorithm upto the Johnson Bound
- Univariate Multiplicity Codes – Decoding upto the List Decoding Capacity
- Bounds on the list size
- Local Decoding (LDC), Local Correction (LCC) of Codes
- Local Correction of Reed Muller Codes
- High Variate Locally correctable/decodable codes
- Local Decoding with constant queries – Matching Vector Codes
- Private Information Retrieval – Definitions, constructions
- Lower Bounds for LDCs – Lower Bound for 2-query/4-query/Kalz-Trevisan/Alrabiah-Guruswami
- Local Testing of Codes:
  - Low-Degree Testing
  - Polischuk-Speilman Test
  - Friedl-Sudan Test
  - Arora-Sudan Test
  - Raz-Safra Test
- Applications: Explicit constructions
  - Combinatorial Designs
  - Subspace Designs
  - Derandomization
  - Hardness vs Randomness

2 Basics of Coding Theory

3 Decoding of Reed-Solomon Codes

4 Multiplicity Codes

Multiplicity codes are a family of recently-introduced algebraic error-correcting codes based on evaluations of polynomials and their derivatives. Specifically, a codeword of a multiplicity code is obtained by evaluating a polynomial of degree at most k, along with all its derivatives of order $< s$, at n points of a finite field $\mathbb{F}_q^m$. These codes were introduced by Kopparty, Saraf and Yekhanin in [KSY14]. Notice that when $s = 1$ this is basically the Reed-Solomon code when $m = 1$ and Reed-Muller code when $m > 1$.

4.1 Construction

Let $s, k, m \in \mathbb{Z}_0$ and let q be a prime power. Let $\Sigma = \mathbb{F}_q^{\binom{s+m-1}{m}}$. For $P(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ we define the order s evaluations of P at $\mathbf{a} \in \mathbb{F}_q$ to be the vector $(P^{(\mathbf{i})}(\mathbf{a}))_{w(\mathbf{i}<s)} \in \Sigma$ where $wt(\mathbf{i}) = \sum_{j=1}^{m} i_j$. Let E be a subset of n points in $\mathbb{F}_q^m$.

Definition 4.1.1: Multiplicity Codes

The multiplicity code of order-s evaluations of degree k polynomials in m variables over all points in E^m is the code over alphabet Σ, and has length n and for each polynomial $P(\boldsymbol{X}) \in \mathbb{F}_q[\boldsymbol{X}]$ with $\deg(P) \leq k$ the corresponding codeword is

$$\mathrm{Enc}_{s,k,m,q}(P) = (P^{(<s)}(\mathbf{a}))_{\mathbf{a}\in E} \in \Sigma^{n^m}$$

Our current interest is in the case $m = 1$. So

$$\mathrm{Enc}_{s,k,1}(P) = \left(\begin{bmatrix} f(a_1) \\ f^{(1)}(a_1) \\ \vdots \\ f^{(s-1)}(a_1) \end{bmatrix}, \begin{bmatrix} f(a_2) \\ f^{(1)}(a_2) \\ \vdots \\ f^{(s-1)}(a_2) \end{bmatrix}, \cdots, \begin{bmatrix} f(a_n) \\ f^{(1)}(a_n) \\ \vdots \\ f^{(s-1)}(a_n) \end{bmatrix} \right)$$

Remark: The above encoding in not the encoding

$$\mathrm{Enc}_{s,k,1} = \left(f(a_1), f^{(1)}(a_1), \cdots, f^{(s-1)}(a_1), f(a_2), f^{(1)}(a_2), \cdots, f^{(s-1)}(a_2), \cdots, f(a_n), f^{(1)}(a_n), \cdots, f^{(s-1)}(a_n) \right)$$

Each alphabet of the codeword is a vector of size s. The same holds for the multivariate case

The above operation of treating a vector as a single alphabet is called *folding*.

4.2 Rate and Distance

We will now calculate the rate and the distance of the code. The block length is n^m. Since we are evaluating all the derivatives of order $< s$, the alphabet size is $q^{\binom{s+m-1}{m}}$. So the number of codewords is $\left(q^{\binom{s+m-1}{m}}\right)^{n^m} = q^{n^m\binom{s+m-1}{m}}$. The number of polynomials in m variables of degree at most k is $q^{\binom{k+m}{m}}$. So the rate of the code is

$$R = \frac{\binom{k+m}{m}}{n^m\binom{s+m-1}{m}} \approx \left(\frac{k}{ns}\right)^m$$

Now using the Multiplicity Schwartz-Zippel lemma we can calculate the distance of the code. We have the relative distance to be $\delta = 1 - \frac{k}{ns}$.

Theorem 4.2.1

The rate and the distance of the multiplicity code are $R = \frac{\binom{k+m}{m}}{n^m\binom{s+m-1}{m}} \approx \left(\frac{k}{ns}\right)^m$ and $\delta = 1 - \frac{k}{ns}$ respectively.

We usually think m and s to be large constant. So as multiplicity code achieves the Singleton bound asymptotically.

5 List Decoding of Univariate Multiplicity Codes

Since we are interested in univariate multiplicity codes, we will set $m = 1$. So we have three parameters k, s, n and the field size q. Therefore, as we have calculated before the rate and distance of the univariate multiplicity code are $R = \frac{k+1}{ns} \approx \frac{k}{sn}$ and $\delta = 1 - \frac{k}{ns}$ respectively.

Theorem 5.1 [Kop15, GW11]
For every $\epsilon \in (0,1)$, there exists $s_0 \approx \frac{1}{\epsilon^2}$ such that the univariate multiplicity code with multiplicity parameter $s > s_0$ can be efficiently list decodable from $\left(1 - \frac{k}{ns} - \epsilon\right)$ fraction of errors.

We will give the proof in [GW11]. It uses polynomial method based arguments. This proof has two steps.

Step 1: Interpolation

Step 2: Reconstruction of close enough codewords

So assume the received word is $w = \left(\alpha_0, \beta_{i,0}, \beta_{i,1}, \ldots, \beta_{i,s-1}\right)_{i=1}^{n}$ and also consider the parameter $m = \sqrt{s} \approx \frac{1}{\epsilon}$. With this we will show the proof of the above theorem.

Proof: In step 1 we will look for an $m+1$ variate polynomial $Q(X, Y_1, \ldots, Y_m)$ which is linear in Y_i's i.e.

$$Q(X, Y_1, \ldots, Y_m) = A_0(X) + A_1(X)Y_1 + \cdots + A_m(X)Y_m$$

And Q follows the following properties:

- $\deg(A_i) \leq D$
- For all $i \in [n]$, Q satisfies some interpolation conditions which we will define now.

Let f is a close enough codeword to w. Define

$$R_f(X) \triangleq Q(X, f(X), f^{(1)}(X), \ldots, f^{(m)}(X))$$

Claim 5.2
If f is a close enough codeword then $R_f(X) \equiv 0$

Claim 5.3
If f and the received word agree on coordinate (i) then $R_f(X)$ has a zero of multiplicity at least $s-1-m$ at α_i.

Proof: Since f agrees with Q at coordinate (j),

$$R_f(\alpha_i) = Q(\alpha_i, f(\alpha_i), f^{(1)}(\alpha_i), \ldots, f^{(m)}(\alpha_i)) = Q(\alpha_i, \beta_{i,0}, \beta_{i,1}, \ldots, \beta_{i,m}) = 0$$

Now

$$R_f^{(1)}(X) = \frac{d}{dX}\left(\sum_{i=0}^{m} A_i(X) f^{(i)}(X)\right) = \sum_{i=0}^{m} \frac{dA_i}{dX} \cdot f^{(i)} + A_i \cdot f^{(i+1)}$$

■

■

Theorem 5.4 [KRZSW18]

The list size above is of constant size only depends on ϵ and independent of the block length.

6 References

[GW11] Venkatesan Guruswami and Carol Wang. *Optimal Rate List Decoding via Derivative Codes*, pages 593–604. Springer Berlin Heidelberg, 2011.

[Kop15] Swastik Kopparty. List-Decoding Multiplicity Codes. *Theory of Computing*, 11(1):149–182, 2015.

[KRZSW18] Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved Decoding of Folded Reed-Solomon and Multiplicity Codes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 212–223. IEEE, October 2018.

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