ICT Presentation

Project Leader : Charchit Gupta(2019102034)
Lokesh Gautham(2019102022)
Sanjai Kumaran(2019112012)

Introduction

The growth of the amount of stored data make the loss of information due to node failures a major problem. To obtain a reliable storage, when a node fails we want to be able to recover the data it contains by using information from the other nodes. This is called the *repair problem*.

Naive method : replicate the same information in different nodes.
More clever strategy : protecting the data by using error-correcting codes.

Examples : Google and Facebook, that use Reed–Solomon (RS) codes in their storage systems.

The (14, 10) Reed-Solomon code, which requires only 40% overhead compared to the (x-1)*100% overhead associated with x-fold replication.

Motivation

There was a optimal function which was made with linear size but restricted length.

Another function was made with field size of $O(2^n)$ and length can be chosen.

This paper combines both to give flexibility on choosing (n,k,r) with a reasonable field size.

LRC code

The most common scenario of failure in modern technologies is the failure of a single node.

A code with properties (n,k,r) where n is the no of bits transferred, k is the no of information bits and r being the no of symbols necessary to recover a symbol is known as a LRC code.

Use of reed solomon code for a single node seems inefficient.

What is locality?

We say that a code C has locality r when every symbol of code x in C can be recovered from a subset of r other symbols of x.
That is when every symbol can be written as a function of r other symbols it is said to have a locality r.
Given a $\in F_q$ conser set of codewords $\mathcal{C}(i,a) = \{x \in \mathcal{C} : x_i = a\}, \quad i \in [n].$
Code is said to have locality r if every i $\in$ [n] there exists a subset Ii $\subset$ [n]\i,|Ii| $\leq$ r such that the restrictions of the sets C(i, a) to the coordinates in Ii for different a are disjoint: $\mathcal{C}_{I_i}(i,a) \cap \mathcal{C}_{I_i}(i,a') = \emptyset, \quad a \neq a'.$
The code $\mathcal{C}_{I_i \cup \{i\}}$ is called a local code of the code C

Preliminaries

Let C be an (n, k,r) LRC code of cardinality q^k over an alphabet of size q, then:
The rate of C satisfies

$$\frac{k}{n} \leq \frac{r}{r+1}$$

The minimum distance of C,

$$d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

CODE CONSTRUCTION

What we want : to construct optimal linear (n, k,r) LRC codes over a finite field alphabet of size q.

q : a prime power >= n.

Assumptions:

1) k is divisible by r
2) n is divisible by r + 1

These restrictions will removed later on.

Method 1: General Construction

Step 1: To find a *good* polynomial, $g(x) \in F_q[x]$

A good polynomial follows the following properties:

1) The degree of g is r + 1,

2) There exists a partition $A = \{A_1, A_2, ..., A_{(n/(r+1))}\}$ of a set $A \subseteq F_q$ of size n into sets of size r + 1, such that g is constant on each set A_i in the partition. Namely for all i = 1,..., n/(r + 1), and any $\alpha, \beta \in A_i$, $g(\alpha) = g(\beta)$.

Method 1: General Construction

Step 2: Define message vector $a \in F_q^k$ as

$a = (a_{0,0}, a_{0,1}, \ldots, a_{0,k/r-1}, a_{1,0}, a_{1,1}, a_{1,2}, \ldots, a_{1,k/r-1}, \ldots, a_{r-1,0}, a_{r-1,1}, \ldots, a_{r-1,k/r-1})$

i.e., $a = (a_{ij},\ i = 0,\ldots,r-1;\ j = 0,\ldots,k/r-1)$

Method 1: General Construction

Step 3: Define the encoding polynomial

$$f_a(x) = \sum_{i=0}^{r-1} f_i(x) x^i$$

Where

$$f_i(x) = \sum_{j=0}^{\frac{k}{r}-1} a_{ij}\, g(x)^j, \quad i = 0, \ldots, r-1$$

$f_i(x)$: *coefficient polynomial*

x : x∈ A

Method 1: General Construction

Step 4: Find the codeword and hence the code.

codeword for a, $c_a = (f_a(x) : x \in A)$

Similarly, the (n,k,r) LRC code C is defined

as the set of n-dimensional vectors $C = \{(f_a(\alpha), \alpha \in A) : a \in F_k^q \}$.

We call the elements of the set A locations and the elements of the vector ($f_a(\alpha)$) *symbols* of the codeword.

Recovery of an Erased Symbol

Suppose that a single symbol is erased from the codeword.

Our task : to recover back the erased symbol corresponding to the location α.

Step 1: Find the set to which α belongs.

i.e., $\alpha \in A_j$, find A_j.

Recovery of an Erased Symbol

Step 2: Find the decoding polynomial for c_α. (quite intuitive)

$$\delta(x) = \sum_{\beta \in A_j \setminus \alpha} c_\beta \prod_{\beta' \in A_j \setminus \{\alpha, \beta\}} \frac{x - \beta'}{\beta - \beta'}$$

Where,

(c_β, $\beta \in A_j \setminus \alpha$) : the remaining r symbols in the locations of the set A_j.

Also, $\delta(\beta) = c_\beta$ for all $\beta \in A_j \setminus \alpha$

Recovery of an Erased Symbol

Step 3: Find the erased symbol.

$c_{\alpha} = \delta(x)|_{x=\alpha}$

Why does it work ?

We know : $c_\alpha = f_a(\alpha)$

Define another decoding polynomial,

$$\partial(x) = \sum_{i=0}^{r-1} f_i(\alpha) x^i$$

It is clear that : $\partial(\alpha) = f_a(\alpha)$

Each $f_i(x)$ is a linear combination of powers of g, therefore it is also constant on the set A_j , i.e., **for any $\beta \in A_j$** and any coefficient polynomial f_i,i=1,...,r-1 **$f_i(\beta)=f_i(\alpha)$**.

$$\partial(\beta) = \sum_{i=0}^{r-1} f_i(\alpha)\beta^i = \sum_{i=0}^{r-1} f_i(\beta)\beta^i = f_a(\beta)$$

Important observation: $\partial(x)$ and $\delta(x)$ are both of degree r-1 and are equal at r points, therefore they must be same.

Hence, $c_\alpha = \delta(\alpha)$ $\quad\quad$ $(\delta(\alpha) = \partial(\alpha) = f_a(\alpha) = c_\alpha)$

What's optimal and Why optimal ?

We know that the minimum distance of an (n,k,r) LRC code satisfies

$$d \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2$$

A code that achieves the bound on the distance with equality is called an optimal LRC code.

For i = 0,...,r − 1; j = 0,..., k/r−1 the k polynomials $g(x)^j x^i$ all are of distinct degrees, and therefore are linearly independent over F, i.e., the mapping $a \rightarrow f_a(x)$ is injective.

The degree of the polynomial $f_a(x)$ is at most: $\left(\frac{k}{r} - 1\right)(r+1) + r - 1 = k + \frac{k}{r} - 2 \leq n - 2$

So, two distinct encoding polynomials f_a and f_b give rise to two distinct code vectors (as they should, a and b : 2 distinct message vectors), hence the dimension of the code is k.

What's optimal and Why optimal ?

No of zeros in the codeword (coresp to message vector a) = No of roots of $f_a(x)$

Max no of zeros in any codeword <= $\max_{f_a, a \in \mathbb{F}_q^k} \deg(f_a)$

Minimum no. of non-zero symbols in a codeword = Minimum distance of the code (encoding is linear)

$$d(\mathcal{C}) \geq n - \max_{f_a, a \in \mathbb{F}_q^k} \deg(f_a) = n - k - \frac{k}{r} + 2$$

Hence, $d_{min} = n - k - k/r + 2$

Example

Parameters: $n = 9$, $k = 4$, $r = 2$, $q = 13$;

Set of points: $A = \{P_1, \ldots, P_9\} \subset F_{13}$

$A = \{A_1 = (1, 3, 9), A_2 = (2, 6, 5), A_3 = (4, 12, 10)\}$

Consider the polynomial $g(x) = x^3$, g is constant on the sets A_i.

Let $a = (a_{0,0}, a_{0,1}, a_{1,0}, a_{1,1})$

$f_a(x) = (a_{0,0} + a_{0,1}g(x)) + x(a_{1,0} + a_{1,1}g(x)) = (a_{0,0} + a_{0,1}x^3) + x(a_{1,0} + a_{1,1}x^3)$

$f_a(x) = a_{0,0} + a_{1,0}x + a_{0,1}x^3 + a_{1,1}x^4$

Example

Say, $\underline{a}_1 = (1, 1, 1, 1)$

Corresponding codeword, $\underline{c}_1 = (f_a(1), f_a(3), f_a(9), f_a(2), f_a(6), f_a(5), f_a(4), f_a(12), f_a(10))$

$\underline{c}_1 = (4, 8, 7, 1, 11, 2, 0, 0, 0)$.

Suppose that the value $f_a(1)$ is erased.

By our construction, it can be recovered by : 2 other codeword symbols, i.e., symbols at locations corresponding to 3 and 9.

The decoding poly : $\delta(x) = 2x + 2$

$\delta(1) = 4$

Example

More *general* message vector, $a = (a_0, a_3, a_1, a_4)$

$f_a(x) = a_0 + a_1x + a_3x^3 + a_4x^4$

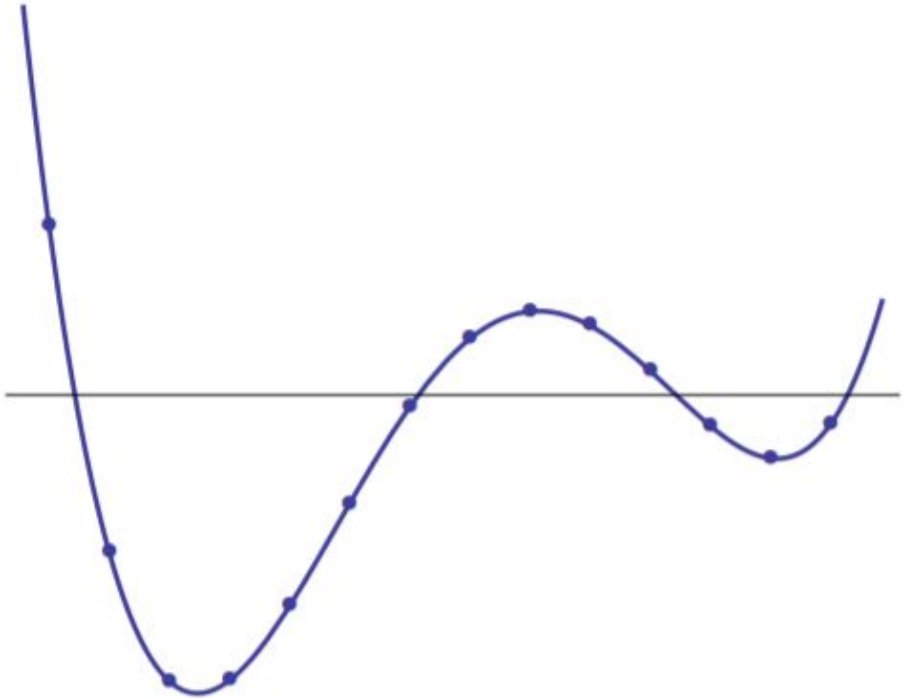
Say any symbol from A_1 is erased,

$f_a(x)|_{A1} = a_0 + a_3 + (a_1 + a_4)x$ (= 2 + 2x for our prev example)

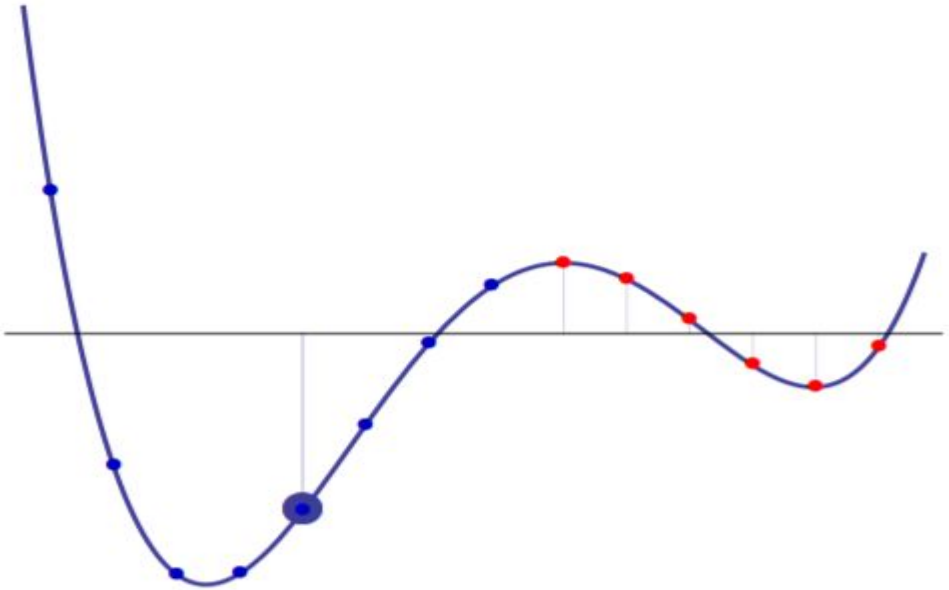
Similarly if any symbol from A_2 is erased,

$f_a(x)|_{A2} = a_0 + 8a_3 + (a_1 + 8a_4)x$

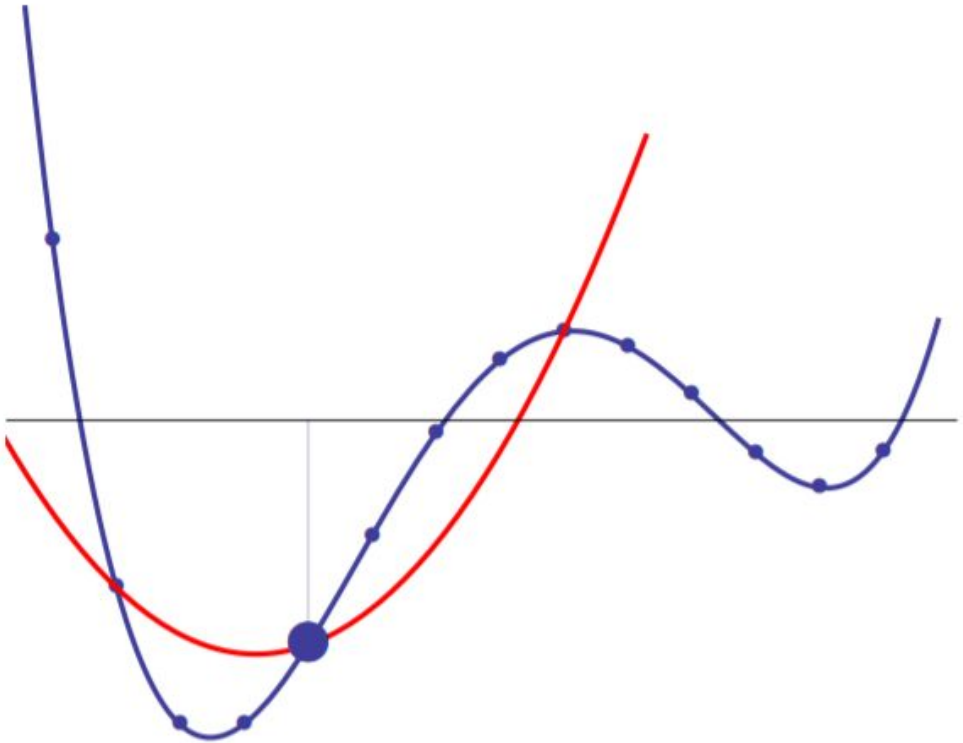
Reed solomon code graph



Reed solomon reconstruction



Reconstruction with locality code



LRC codes with multiple recovering sets

This is an extension of original LRC.

We will have more than one sets to recover a current

We assume the recovering sets to be disjoint

Application : In distributed storage applications

- subsets of data are accessed more frequent that the rest (hot data)
- Therefore, using this kind of construction, we can make multiple users access this data without delays

Formal definition

Let $\mathcal{C}$ be called a LRC (t) code $\{t$ recovery sets$\}$

Let $\mathcal{C} \in \mathbb{F}^n$ ($\mathbb{F}$ is the field.)

Then, for every i in $[0, n-1]$, there exists t subsets of sizes $r_1, r_2, \ldots r_t$ each, such that any symbol is a function of the values individually in those subsets.

This can be written as an $(n, k, \{r_1, r_2 \ldots r_t\})$ LRC code.

Methods of construction

There are primarily two methods to construct multiple recovering LRC codes

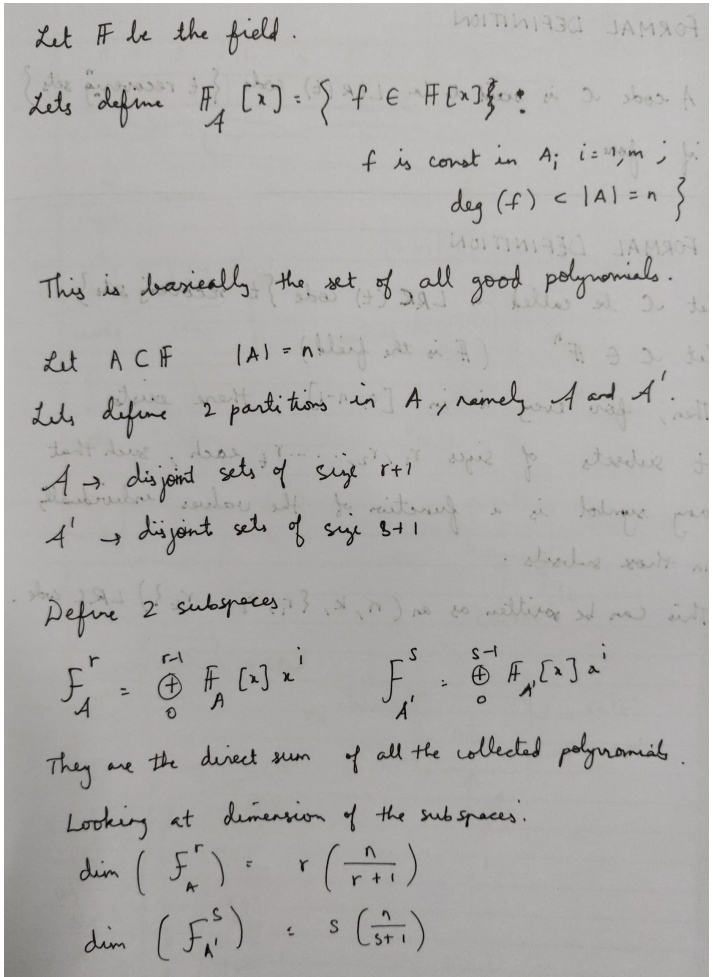
1. Algebraic LRC Codes With Multiple Recovering Sets
2. LRC Product Codes

Systematic encoding of LRC codes

$$\phi_{i,j}(\beta_{i,l}) = \delta_{j,l}.$$

$$f_a(x) = \sum_{i=1}^{k/r} f_i(x)\left(\sum_{j=1}^{r} a_{i,j}\phi_{i,j}(x)\right).$$

Algebraic LRC Codes With Multiple Recovering Sets



Let $\mathbb{F}$ be the field.

Lets define $\mathbb{F}_A[x] = \{ f \in \mathbb{F}[x] :$

f is const in A_i; $i = \overline{1,m}$;

$\deg(f) < |A| = n \}$

This is basically the set of all good polynomials.

Let $A \subset \mathbb{F}$ $\quad |A| = n$

Lets define 2 partitions in A, namely A and A'.

$A \rightarrow$ disjoint sets of size $r+1$

$A' \rightarrow$ disjoint sets of size $s+1$

Define 2 subspaces

$$F_A^r = \bigoplus_0^{r-1} \mathbb{F}_A[x]\, x^i \qquad F_{A'}^s = \bigoplus_0^{s-1} \mathbb{F}_{A'}[x]\, a^i$$

They are the direct sum of all the collected polynomials.

Looking at dimension of the subspaces:

$$\dim\left(F_A^r\right) = r\left(\frac{n}{r+1}\right)$$

$$\dim\left(F_{A'}^s\right) = s\left(\frac{n}{s+1}\right)$$

For an integer m, let P_m be defined as set of all polynomials of deg $\leq m$.

Now, defining $V_m = F_A^r \cap F_{A'}^s \cap P_m$

Construction.

Assuming $\dim\left(F_A^r \cap F_{A'}^s\right) \geq k$,

choose m (smallest) s.t $\dim(V_m) = k$.

Define a mapping $\phi : \mathbb{F}^k \to V_m$.

(linear and injective)

Since linear,

$$\phi(a) = \sum_{i=0}^{k-1} a_i \, g_i(x)$$

This $\phi(a)$ can be denoted as $f_a(x)$.

$f_a(x)$ is the encoding polynomial.

Orthogonal partitions: if A and B are orthogonal, the $|X \cap Y| \leq 1$ where $X \in A$ and $Y \in B$.

Here, if A_1 and A_2 are orthogonal, then every symbol in code has 2 distinct recovering sets of size r and s respectively.

Proposition

consider a finite field F_q such that q-1 is not prime. Let r and s be 2 factors of q-1 which are coprime. Then there exists a LRC(q) code of length q-1 over F_q such that each symbol has 2 disjoint recovering sets of size r-1 and s-1.

LRC product code

- Another way of construction LRC(t) codes is to take t LRC codes and take product of the corresponding subspaces

For simplicity, we are considering construction of LRC(2) code construction :

considering 2 LRC codes with {n1, k1, r1} and {n2, k2, r2} we can construct a (n1n2, k1k2, {r1, r2}) LRC code.

Construction

Let’s consider C1 and C2 to be linear and define a linear mapping Φ such that

$$\Phi = \Phi_1 \otimes \Phi_2 : \mathbb{F}^{k_1 k_2} \rightarrow \oplus_{i=0}^{r_1 - 1} \mathbb{F}_{\mathcal{A}_1}[x] x^i \otimes \oplus_{j=0}^{r_2 - 1} \mathbb{F}_{\mathcal{A}_2}[y] y^j$$

This is the tensor product of individual mappings.

We can define the encoding polynomial as $f_a(x, y) = \Phi(a).$

Let C be a code constructed by above mentioned construction.

Considering a set pair $A_1 x A_2$

If (x_0,y_0) is a point in $A_1 x A_2$, then value of $f_a(x_0,y_0)$ can be evaluated by accessing r_1 and r_2 respectively.

This is similar to superposition.

We also find the distance of product to satisfy $d=d_1 d_2$

Comparison

Here we will compare both methods to construct LRC codes with multiple recovering sets

Algebraic construction	LRC product code
Considering a fixed target code rate if we find the minimum distance of the code it is generally higher than LRC product code.	Has less minimum for a fixed code rate than algebraic construction.
Less flexible.	More flexible as it gives more disjoint recovering sets by design.
More complex as we have to construct several mutually orthogonal partitions along with their good polynomials which is a complex task.	Less complex as we only do product of subspaces
Construction requires field size of order n.	Construction requires field size of order $\sqrt{n}$

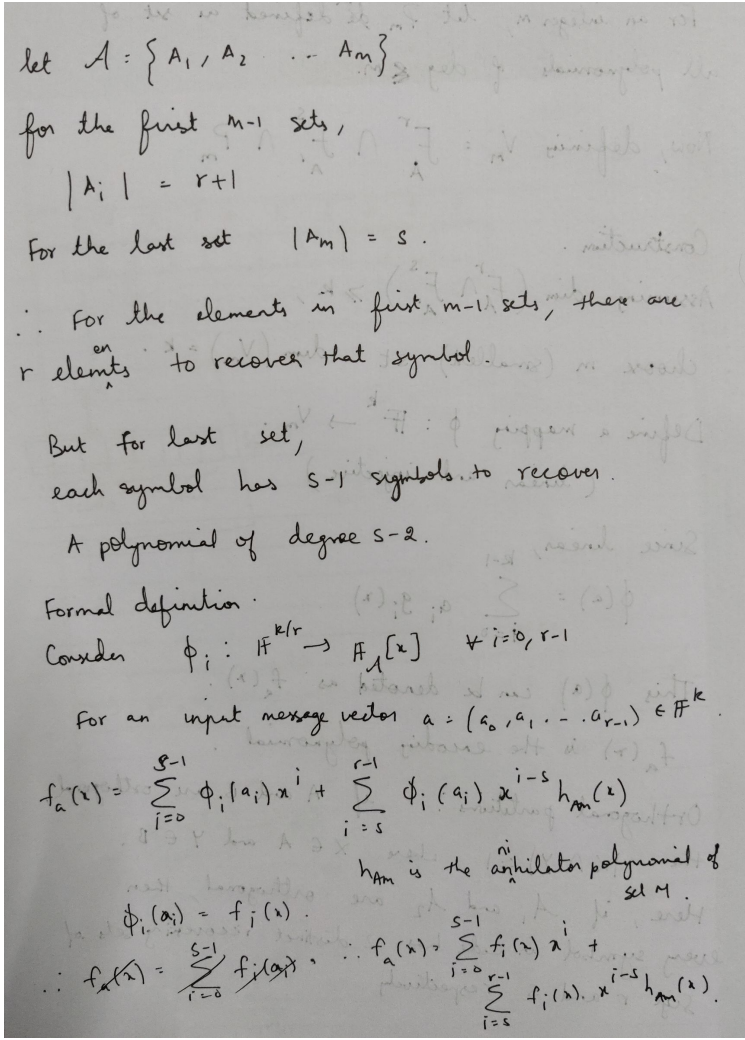
Generalization

Earlier we assumed n to be divisible by r+1 so that we can make complete partitions.we consider a general n such that n mod (r+1) is not 0 and construct a code for this.

Construction

Let us consider n mod (r+1) is s which is not equal to 1.

Let us consider m as ceil(n/r +1) A



This type of construction gives us a (n,k,r) LRC code with

$$d_{min} \geq n - k - \left\lceil \frac{k}{r} \right\rceil + 1. \qquad -(1)$$

We already know $d_{min} \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2$.

Optimal LRC codes satisfy this bound.

But for arbitrary length LRC,

the d_{min} decreases atmost by $1 - [\text{by } (1)]$

This can be proved by looking at degree of encoding polynomial.

which is at most $\left(\frac{k+1}{r} - 1 \right)(r+1) + (r-1)$

Simplifying, we get

$d_{min} \not\leq n$

$$d_{min} \geq n - k - \left\lceil \frac{k}{r} \right\rceil + 1.$$

This is so powerful as irrespective of n, the d_{min} reduces at most by 1.

Conclusion

Generalization of reed solomon codes taking locality into account.

Extended the main construction to codes with multiple independent recovering sets so that the lost symbol can be corrected by accessing several different r-subsets of the codeword coordinates.