Introduction to Error Correcting Codes

Amir Shpilka
Arazim ©

February 4, 2016

1 Concatenated RS-code

Reminder 1. An RS code is a $[n_1, k_1, d_1]_{q=2^l}$ where $k_1 + d_1 = n_1$.
Code $[c_2, l, d_2]_2$
An RS∘ code is $[n_1 \cdot n_2, k \cdot l, d_1 \cdot d_2]_2$ since $(\mathbb{F}_2)^l \cong \mathbb{F}_{2^l}$. We will show an algorithm that corrects code with a number of errors that is $e < \frac{d_1 \cdot d_2}{2}$.

As a warm-up: fixing errors with $e < \frac{d_1 d_2}{4}$ errors in a time of $poly\left(n_1, 2^l \cdot n_2\right)$ (for $q = \mathcal{O}(n_1)$ we have $poly(n_1)$).
The algorithm:

1. We will fix each one of the n_1 blocks, as good as possible (Brute-force).
2. Now, we will fix the output of that using W-B for RS and arrive at the code word that we sent.

Analysis: If the block is "small" meaning that there were at most $\frac{d_2}{2}$ errors, then step 1 would have fixed it. Therefore, since there were less than $\frac{d_1}{2} \cdot \frac{d_2}{2}$ mistakes, the number of blocks where we will have a mistake is smaller than $\frac{d_1}{2}$
And now, the W-B algorithm fixes this without any errors.

1.1 Some markings

$\bar{x} = \left(x_1, \ldots, x_{n_1}\right) \in \{0,1\}^{n_1 \cdot n_2}$ is a code word, where $x_i \in \{0,1\}^{n_2}$.
The codeword after the interference (e.g. with noise) is marked as $\bar{y} = \left(y_1, \ldots, y_{n_1}\right)$ and in particular we have $dist\left(\bar{y}, \bar{x}\right)$
We will denote the result of the first stage of the algorithm as $\left(u_1, \ldots, u_{n_1}\right)$. $u_i = \arg_{u\in} \min_{code} dist(u_i, y_i)$
We will mark $e_i' = dist(u_i, y_i)$ and $\hat{e_i}' = dist(x_i, y_i)$.

$$\sum e_i' \leq \sum e_i < \frac{d_1 \cdot d_2}{2}$$

Now, for all i we will define $p_i = \min\left\{1, \frac{e_i'}{d_{2/2}}\right\}$ and we will define v_i in the following way:

$$v_i = \begin{cases} ? & \text{With probability } p_i \\ u_i & \text{With probability } 1 - p_i \end{cases}$$

Claim 1. $\Pr\left[v_i = ?\right] + 2\Pr\left[\text{mistake at coordinate } i\right] < \frac{e_i}{d_{2/2}}$.

From the claim we have $\mathbb{E}\left[\#?\right] + 2 \cdot \mathbb{E}\left[\#errors\right] < \frac{\sum e_i}{d_{2/2}} < d_1$ Thus, $\mathbb{E}\left[\#? + 2 \cdot \#errors\right] < d_1$.

proof of the claim. Splitting into cases:

1. $u_i = x_i$ then
$$\Pr\left[v_i = ?\right] = \min\left\{1, \frac{e_i}{d_2/2}\right\} \le \frac{e_i}{d_2/2}$$
$$\Pr\left[\begin{array}{c} v_i \neq ? \\ n_i \neq x_1 \end{array}\right] = 0$$

2. If $u_i \neq x_i$ then $e_i + e'_i \ge dist\left(u_i, x_i\right) \ge d_2$ we need to calculate
$$\overbrace{\Pr\left[v_i = ?\right]}^{p_i} + 2 \cdot \overbrace{\Pr\left[\begin{array}{c} v_i \neq ? \\ n_i \neq x_1 \end{array}\right]}^{1-p_i} = p_i + 2\left(1 - p_i\right) = 2 - p_i = 1 + \left(1 - p_i\right) = 2 - p_i$$
If $p_i = 1$ then $2 - p_i = 1 \le \frac{e_i}{d_2/2}$
On the other hand, if $p_i = \frac{e_i}{d_2/2}$ then $2 - p_i \le 2 - 2\frac{d_2 - e_i}{d_2} - \frac{2e_i}{d_2}$.

□

All of this happens in the expected value, we want this to occur always. We will choose a $p \in [0,1]$ according to the uniform distribution. For the i-th coordinate we will define

$$v_i = \begin{cases} ? & p \le \frac{e'_i}{d_2/2} \\ u_i & else \end{cases}$$

Noticing that $\Pr\left[v_i = ?\right] = \min\left\{1, \frac{e'_i}{d_2/2}\right\}$ then we still have

$$\mathbb{E}\left[\#? + 2 \cdot \#errors\right] < d_1$$

Also, the "interesting" values of p are the values where the vector $\bar{v}$ changes, they are in the set $\{0,1\} \cup \left\{{e'_i}/{d_2/2}\right\}_{i=1,\ldots,n_1}$. Thus, we arrive at the following algorithm

---

Algorithm 1 Random RS∘ decoder

1: For all y_i find the closest codeword and mark it with u_i we will
2: Define $e'_i = dist(u_i, y_i)$.
3: Choose $p \in \{0,1\} \cup \left\{{e'_i}/{d_2/2}\right\}_{i=1,\ldots,n_1}$:
4: Define $\bar{v}\left(p\right)$ as follows
$$v_i\left(p\right) = \begin{cases} ? & p \le {e'_i}/{d_2/2} \\ u_i & else \end{cases}$$
5: Using the WB algorithm, fix $\bar{v}\left(p\right)$.
6: **if** $\mathrm{dist}(\bar{v}(p), \bar{y}) < \frac{d_1 \cdot d_2}{2}$ **then** Done.
7: **else**
8:     **goto** 3
9: **end if**
10: if the distance from the returned word is smaller than $\frac{d_1 d_2}{2}$ we are done. (if not then we'll move on to the next p.)

---

2 List-decoding

Up until now we've dealt with unique decoding, meaning that if the number of errors is at most $\frac{d-1}{2}$, we will return the closest codeword. Now we will discuss a system the returns more than one matching codeword.

Definition 1. We say that a code $C \subseteq \mathbb{F}_q^n$ is a (e,l) code if in every ball with a radius of e there is at most l codewords.

Example 1. If $dist(C) = d$ then C is a $\left(\frac{d-1}{2}, 1\right)$-code.

Our goal here is given a (e,l) code and a word $\bar{y}$ to find in an efficient way all of the codewords at a distance of at most e from $\bar{y}$.
Such an algorithm is called a List-Decoding algorithm.

2.1 Conversion from an n,k,d code

If we assume that C is $[n,k,d]_q$. For what parameters is C an (e,l) code?
Assume that $\bar{x_1}, \bar{x_2}, \ldots, \bar{x_l}$ are code words with distances from $\bar{y}$ of at most e meaning that each $\bar{x_i}$ agrees with $\bar{y}$ in at least $n-e$ coordinates. We will create double sided graph, with all of the $\bar{x_i}$ on one side and $\bar{y}$ on the other and connect each $\bar{x_i}$ to y_j if the j-th coordinate of $\bar{x_i}$ equals to y_j.

How many neighbors in common do $\bar{x_i}$ and $\bar{x_j}$ have $(i \neq j)$? At most $n-d$.

We will want to bound l using the aforementioned bounds.
In this graph we do not have a $K_{n-d+1,2}$ (meaning that every two coordinates have less than $n-d+1$ neighbors) and in our case (RS codes) we have no $K_{k,2}$.
Claim 2. If $t > \sqrt{nk}$ then $l \leq \frac{nt}{t^2-nk}$

In order to bound l we will count the number of objects of the form $\bar{x_i} y_j \bar{x_k}$ meaning that there is an edge between $\bar{x_i}$ and y_j and between y_j and $\bar{x_k}$, denote this as $\#$. Let $\Delta_1, \ldots, \Delta_n$ be the degrees of the edges of L.

$$\# = \sum_{i=0}^{n} \binom{\Delta_i}{2}$$

On the other hand, we also have:

$$\# \leq (k-1)\binom{l}{2}$$

$$\sum \frac{\Delta_i(\Delta_i - 1)}{2} = \# \leq (k-1) \cdot \frac{l(l-1)}{2} \Rightarrow \sum \Delta_i^2 - \sum \Delta_i \leq (k-1)\, l\, (l-1)$$

It is known that $\sum \Delta_i = l \cdot t$, then according to the Cauchy-Schwartz inequality

$$\left(\sum_{n=1}^{n} \Delta_i\right)^2 \leq n \sum \Delta_i^2 \Rightarrow \frac{l^2 t^2}{n} - lt \leq (k-1)\, l\, (l-1)$$

$$\Rightarrow l \cdot \frac{t^2}{n} - t \leq (k-1)(l-1) < k \cdot l \Rightarrow l\left(\frac{t^2}{n} - k\right) \leq t$$

And then, if $t^2 < nk$ we have $l \leq \frac{nt}{t^2-nk}$.

Claim 3. A RS $[n, k, n-k+1]_q$ code is (e, l) for $(n-e)^2 > nk$ and $l \leq \frac{n(n-e)}{(n-e)^2 - nk}$.

Corollary 1. *Let $\bar{y} \in \mathbb{F}_q^n$ $(n < q)$ the number of polynomials with a degree smaller than k for which*

$$\#\left\{i | f\left(\alpha_i\right) = y_i\right\} > \sqrt{nk}$$

is at most n^2.

Corollary 2. *The number of RS codes at a distance of $< n - \sqrt{nk}$ from the word $\bar{y}$ is $\leq n^2$*

In List-decoding we can expect that from a percentage of errors that is smaller than $1 - \sqrt{R}$ we can calculate all of the close codewords. Notice that if we would ask for unique-decoding then the number of errors that we would be able to fix is $\frac{1}{2}(1-R) = \frac{1}{2}\delta$.
Our goal now is to find an efficient List-decoding algorithm for RS.

Problem 1. There are polynomials f_1, f_2 with degrees $< k$ such that for all α_i, $y_i = f_1(\alpha_i)$ or $y_i = f_2(\alpha_i)$. We need to find f_1 and f_2 (under the assumption that there are enough agreements with each of them).

We will notice that at every point α_i we have

$$0 = \left(y_i - f_1(\alpha_i)\right)\left(y_i - f_2\left(\alpha_i\right)\right)$$

and

$$0 = y_i^2 - y_i \overbrace{\left(f_1\left(\alpha_1\right) + f_2\left(\alpha_i\right)\right)}^{B(\alpha_i)} + \overbrace{f_1\left(\alpha_i\right) \cdot f_2\left(\alpha_i\right)}^{C(\alpha_i)}$$

Where $B = f_1 + f_2$ and $C = f_1 \cdot f_2$, $\deg B < k$ and $\deg C < 2k-1$. We will solve the following set of equations with variables that are the coefficients of B and C. For $i \in \{1, 2, \ldots, n\}$ $y_i^2 - y_i \cdot B\left(\alpha_i\right) + C\left(\alpha_i\right) = 0$. It is known that this system has a solution. Let us assume that we have solved it and arrived at $B'(x), C'(x)$ then we get a polynomial $y^2 - y \cdot B'(x) + C'(x)$ we will split the polynomial into irreducible factors (if possible) .

Claim 4. If the number of times that f_1 agrees with $\bar{y}$ is at least $2k+1$ then $y - f_1(x)$ is one the factors.

Proof. Let α_i for which $f_1(\alpha_i) = y_i$ be an α_i for which this is true $y_i^2 - y_i B'(\alpha_i) + C'(\alpha_i) = 0$ and thus $f_i(\alpha_i)^2 - f_i(\alpha_i) B'(\alpha_i) + C'(\alpha_i) = 0$. In other words, at α_i the polynomial $\overbrace{f_1(x)^2}^{\deg < 2k-1} + \overbrace{f_1(x) B'(x)}^{<2k-1} + \overbrace{C'(x)}^{<2k-1}$ vanishes. In particular the polynomial is zero meaning that it is one of the factors. ∎