Introduction to Error Correcting Codes

Amir Shpilka
Arazim ©

February 4, 2016

In this lesson we will be continuing List-decoding algorithms for RS codes and show more bounds for codes. Continuing from last lesson, where we showed that there are at most n^2 polynomials with a degree $< k$ which agree with a word on at least $\sqrt{nk}$ coordinates.
In this lesson we will show two algorithms, the first will find all of the RS codewords (polynomials with a degree $< k$) which:

- Agree on at least $2\sqrt{nk}$ coordinates of a given problem.
- Agree on at least $\sqrt{2}\sqrt{nk}$ coordinates of a given problem.

Reminder 1. In the "toy problem" we found a polynomial $Q(y,x) = y^2 + B(x)y + C(x)$ such that if f has a degree $< k$ which agrees with the word on more than $2k$ coordinates then $y - f(x) | Q(x,y)$ (we will show this by showing that $Q\left(x, f(x)\right) = 0$[1]).
The idea for the algorithm today:
We will find a polynomial $Q(x,y)$ such that if $(y_1, y_2, \ldots, y_n)$ is the word we are trying to fix then $Q(\alpha_i, y_i) = 0$.
On the other hand if $\mathrm{Enc}(f)$ agrees with $\bar{y}$ on "many" coordinates then $y - f(x) | Q(x,y)$
We can think about Q as a polynomial with the variable y and over the field $\mathbb{F}(x)$ where

$$\mathbb{F}(x) = \left\{ \frac{a(x)}{b(x)} : a, b \text{ are polynomials and } b \neq 0 \right\}$$

This is called the rational function field over x.

1 The first algorithm

Let $Q(x,y)$ be a polynomial with a degree smaller or equal to d_x on x and d_y on y, meaning that if $x^i \cdot y^j$ is in Q then $i \leq d_x, j \leq d_y$. What we will want to occur is that

1. There are "enough" coefficients such that we can find a $Q \not\equiv 0$ for which $\forall i. Q(\alpha_i, y_i) = 0$. In particular, we need that $d_x \cdot d_y > n$ since $d_x \cdot d_y$ is the number of monomials.
2. $\deg Q\left(x, f(x)\right) < 2\sqrt{nk}$ where $\deg f(x) < k$,

$$\deg\Big(Q\left(x, f(x)\right)\Big) \leq (d_x - 1) + \big(d_y - 1\big)(k - 1)$$

   Since $d_x - 1$ is the highest power of x allowed, $d_y - 1$ is the highest power of y allowed and $k - 1$ is the highest power in $f(x)$. A good solution for this is $d_x = \left\lceil \sqrt{nk} \right\rceil$ and $d_y = \left\lceil \sqrt{n/k} \right\rceil$ For such a choice of d_x and d_y there exists a $Q(x,y)$ with a degree in x that is smaller than d_x and a degree over y that is smaller than d_y such that $\forall i. Q(\alpha_i, y_i) = 0$.

[1]Since we already know that if $p(y)$ is a polynomial and $p(\alpha) = 0$ then $y - \alpha | p(y)$

Given that, let f be a polynomial with a degree smaller than k such that there are at least $2\sqrt{nk}$ α_i such that $f(\alpha_i) = y_i$. For each point of agreement such as that we have $0 = Q(\alpha_i, y_i) = Q(\alpha_i, f(\alpha_i))$ and in particular, the polynomial $Q(x, f(x))$ vanishes on every agreement point of $\mathrm{Enc}(f)$ and $(y_1, y_2, \ldots, y_n)$ and in particular there are $\geq 2\sqrt{nk}$ zeros. Since $\deg\left(Q(x, f(x))\right) < 2\sqrt{nk}$ we get that $Q(x, f(x)) \equiv 0 \Rightarrow y - f(x) | Q$.
Thus, the algorithm is as follows:

1. We will find a $Q \neq 0$ such that
   (a) $\forall i . Q(\alpha_i, y_i) = 0$
   (b) The x degree is $< \left\lceil \sqrt{nk} \right\rceil$.
   (c) The y degree is $< \left\lceil \sqrt{n/k} \right\rceil$.
2. We will find all of the irreducible factors of Q of the form $y - f(x)$, where $\deg f < k$ and for each of there we will check if it agrees with the word on enough coordinates.

2 The second algorithm

In reality we could've allowed another monomial in A and still arrive at a polynomial for which $Q(\alpha_i, y_i) = 0$ and $\deg\left(Q(x, f(x))\right) < 2\sqrt{nk}$. In order for the second condition to occur we will require for every monomial $x^a y^b$ that the number of agreements (marked as t) is larger than $a + (k-1)b$.
Let us assume that we require t agreements. The total number of monomials we can have is

$$\#\left\{(a, b) : a + (k-1) \cdot b < t\right\}$$

when $b = 0$ we have t such monomials, $b = 1$ we have $t - (k-1)$ and thus for all b up to m we have $t - m(k-1)$ such monomials where $m < \frac{t}{k-1}$. Calculating this sum:

$$t \cdot \frac{t}{k-1} - (k-1) \sum_{i=0}^{\frac{t}{k-1}} i \approx \frac{t^2}{k-1} - \cancel{(k-1)} \frac{1}{2} \frac{t^2}{(k-1)^{\cancel{2}}} = \frac{1}{2} \frac{t^2}{k-1} \overset{?}{>} n$$

If we have this many monomials we will be able to solve the linear system of equations and since $t \geq \sqrt{2nk}$ we have the required inequality. This gives us the following algorithm:

1. Find a $Q(x, y) \not\equiv 0$ which only contains monomials of the type $x^a y^b$ where $a + (k-1)b < \sqrt{2nk}$.
2. $\forall i . Q(\alpha_i, y_i)$ we will find all of the irreducible factors of Q of the type $y - f(x)$ and save those which agree on at least $\sqrt{2nk}$ coordinates.

3 Recap

Returning to what we did at the beginning of the course, we showed the following bounds:

- Singleton: for an $[n, k, d]$ code we have that $k + d \leq n + 1$ and $R + \delta \leq 1$
- Hamming: $R + H\left(\frac{\delta}{2}\right) < 1$

We also showed that there exist codes such that$R > 1 - H(\delta)$ meaning that the bound is tight.

3.1 Plotkin's bound

We showed this in the second homework,

$$2^k < 2\left\lfloor \frac{d}{2d-n} \right\rfloor$$

And in the case of $d = \left(\frac{1}{2}+\varepsilon\right)n$. This gives us $2^k < 2 \cdot \frac{\left(\frac{1}{2}+\varepsilon\right)n}{2\varepsilon n}$ and $2^k < \mathcal{O}\left(\frac{1}{\varepsilon}\right)$. We will now show geometric proof for this.
Using an embedding of $\{0,1\}^n$ in $\mathbb{R}^n$ as follows, $0 \to 1$ and $1 \to -1$, e.g. $\{0,1\}^n \to \{1,-1\}^n$. Let v_1, v_2 be codewords ($\in \{\pm 1\}^n$), then $\langle v_i, v_i \rangle = n$ and for all $i \neq j$ we have

$$\left\langle v_i, v_j \right\rangle = \#\left\{\text{Agreements}\right\} - \#\left\{\text{Disagreements}\right\} \leq n - 2dist\left(v_i, v_j\right)$$

Corollary 1. *A code that contains the words $v_1, v_2, \ldots, v_m$ and the minimal distance is δn gives a which gives a set of m vectors for which $\langle v_i, v_i \rangle = n$ and for all $i \neq j$*

$$\left\langle v_i, v_j \right\rangle \leq n - 2dist\left(v_i, v_j\right) \leq n - 2\delta n$$

If we normalize $\hat{v}_i = \frac{v_i}{\sqrt{n}}, \|\hat{v}_i\| = 1$ and $\left\langle v_i, v_j \right\rangle \leq 1 - 2\delta$. If $\frac{1}{2} + \varepsilon = \delta$ then we have for all $i \neq j$, $\left\langle v_i, v_j \right\rangle \leq -2\varepsilon$

Claim 1. Let $v_1, \ldots, v_m \in \mathbb{R}^n$ such that $\|v_i\| = 1$ and for all $i \neq j$ we have $\left\langle v_i, v_j \right\rangle \leq 0$ then $m \leq 2n$.

Proof. We will assume WLOG that $v_1 = (\theta, 0, \ldots, 0)$, $\theta > 0$ and we will represent each v_i as $v_i = (\alpha_i, - - u_i - -)$, then $0 = \langle v_1, v_i \rangle - \alpha_i \cdot \theta$ and

$$\left\langle u_i, u_j \right\rangle = \overbrace{\left\langle v_i, v_j \right\rangle}^{0\geq} - \alpha_i \cdot \alpha_j \leq 0$$

There can be at most one more vector v_i such that $v_i = (\alpha_i, 0, \ldots, 0)$ (since if there were more then we would have 3 vectors of the type $(\alpha, 0, \ldots, 0)$ and two of them would have a positive inner product) therefore it is sufficient to get $m - 2$ vectors with a dimension of $n - 1$ which hold the conditions. $\square$

Corollary 2. *If $d = \frac{n}{2}$ in an $[n, k, d]$ code then the number of words is smaller or equal to $2n$.*

Lemma 1. *1. Let $\alpha > 0$ and we will assume that there are $v_1, \ldots, v_m \in \mathbb{R}^m$ with $\|v_i\| = 1, \left\langle v_i, v_j \right\rangle \leq -\alpha$ then $m \leq 1 + \frac{1}{\alpha}$.*

2. If $y, v_1, \ldots, v_m \in \mathbb{R}^n$ such that $\langle v_i, y \rangle > 0$ and for all $i \neq j$, $\left\langle v_i, v_j \right\rangle \leq 0$ then $m \leq n$.

proof of 1. Let $u = v_1 + \cdots + v_m$

$$0 \leq \langle u, u \rangle = \sum_i \langle v_i, v_i \rangle + \sum_{i \neq j} \left\langle v_i, v_j \right\rangle \leq m - m\left(m-1\right)\alpha$$

Thus, $\left(m-1\right)\alpha \leq 1$ and $m \leq \frac{1}{\alpha} + 1$ $\square$

proof of 2. We will assume by contradiction that there are such vectors as mentioned previously with $m > n$. In particular, there is a non trivial linear combination

$$\sum_{i=0}^{m} \alpha_i v_i = \vec{0}$$

And,

$$z = \sum_{i:\alpha_i > 0} \alpha_i v_i = \sum_{i:\alpha_i < 0} \alpha_i v_i$$

1. If $z \neq 0$ then $\langle z, z\rangle > 0$

$$0 < \langle z, z\rangle = \left\langle \sum_{i:\alpha_i>0} \alpha_i v_i, -\sum_{i:\alpha_i<0} \alpha_i v_i \right\rangle = \sum_{\substack{i:\alpha_i>0 \\ j:\alpha_j<0}} \alpha_i \cdot (-\alpha_j) \cdot \langle v_i, v_j\rangle \leq 0$$

2. If $z = \vec{0}$

$$0 = \langle y, z\rangle = \left\langle y, \sum_{\alpha_i>0} \alpha_i v_i \right\rangle = \sum_{\alpha_i>0} \alpha_i \langle y, v_i\rangle > 0$$

   and again we have a contradiction.

$\square$

Corollary 3. *If $d = \left(\frac{1}{2} + \varepsilon\right) n$ in an $[n, k, d]_2$ code then the number of code words is smaller or equal to $1 + \frac{1}{2\varepsilon}$*

Theorem 1 (Johnson bound)**.** *Every $[n, k, d]_2$ code is also a $(\tau \cdot n - 1, n)$-code for $\tau = \frac{1}{2}\left(1 - \sqrt{1-2\delta}\right)$*

Proof. Let $\bar{y}$ be a vector and let $v_1, \ldots, v_n$ be codewords in the ball with a radius of $\tau n - 1$ around y. According to what we have said, if we embed the problem in $\mathbb{R}^n$ we will get vectors $v_1, \ldots, v_n$, $\bar{y}$ for which $\|\bar{y}\| = 1$ and for all i we have $\|v_i\| = 1$, for all $i \neq j$, $\langle v_i, v_j\rangle \leq 1 - 2\delta$, $\langle v_i, y\rangle > 1 - 2\tau$.
Let $\lambda > 0$ be a parameter which we will define in the future. We will define $v'_i = v_i - \lambda y$

$$\left\langle v'_i, v'_j \right\rangle = \langle v_i, v_j\rangle - \lambda \langle v_i, y\rangle - \lambda \langle v_j, y\rangle + \lambda^2 \leq 1 - 2\lambda - 2\lambda(1-2\tau) + \lambda^2 = (1-\lambda)^2 - 2\delta + 4\lambda\tau$$

We will take $\lambda = 1 - 2\tau$ and we will get for that λ,

$$\left\langle v'_i, v'_j \right\rangle \leq 4\tau^2 - 2\delta + 4(1-2\tau)\cdot\tau = 4\tau - 4\tau^2 - 2\delta$$

For $\tau = \frac{1}{2}\left(1 - \sqrt{1-2\delta}\right)$ the equation is equal to 0. In addition $\langle v_i, y\rangle = \langle v_i, y\rangle - \lambda > 1 - 2\tau - \lambda = 0$ $\square$