CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 11: 12/9/22

Scribe: Amit Kumar Mallik *Lecturer: Mrinal Kumar*

1 Reed Solomon Codes

A code where codewords are picked at random turns out to be a good with high probability. However, for applications, we need explicit constructions. Reed Solomon codes is one such construction.
We denote Reed Solomon codes as $RS(k, n)$ for some integers $k \leq n$, defined as follows.
Let $\mathbb{F}$ be a field with $|\mathbb{F}| \geqslant n$. We view the messages as coming from the set

$$Msg := \{g(x) \in \mathbb{F}[x] \mid deg(g) \leqslant k-1\}$$

Thus, $|Msg| = |\mathbb{F}|^k$.
Pick some distinct $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}$. Encoding of a message is given as

$$\text{Enc}(g) = (g(\alpha_1), g(\alpha_2), \ldots g(\alpha_n)) \in \mathbb{F}^n.$$

The encodings are called codewords of the code.
Let $g(x) = g_0 + g_1 x + g_2 x^2 + \ldots g_{k-1} x^{k-1}$. Observe that $\text{Enc}(g)$ can also be written as a matrix-vector product as follows

$$\text{Enc}(g) = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_1^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \ldots & \alpha_n^{k-1} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}$$

Notation. For any two codewords $r, s \in \mathbb{F}^n$ in the code the distance between them $\Delta(r, s)$ is the number of positions in which they differ, i.e.,

$$\Delta(r, s) = |\{i : r_i \neq s_i\}|.$$

The normalized distance δ of a code is the minimum distance between any two codewords divided by n.
We now discuss some properties of $RS(k, n)$.

1.1 Properties

1. **Linear Code**
   Observe that $\text{Enc}(g + h) = \text{Enc}(g) + \text{Enc}(h)$.
2. **Rate**
   $Rate = \dfrac{\log_{|\mathbb{F}|} |Msg|}{n} = \dfrac{k}{n}$
3. **Minimum Distance** Observe that $g(\alpha_i) = h(\alpha_i) \iff (g-h)(\alpha_i) = 0$. But, $g - h$ is a polynomial of degree at most $k-1$ and thus has at most $k-1$ roots. So, $\text{Enc}(g)$ and $\text{Enc}(h)$ can match at at most $k-1$ position. Thus, for all g, h $\Delta(\text{Enc}(g), \text{Enc}(h)) \geqslant n - k + 1 \implies \delta \geqslant \dfrac{n-k+1}{n}$

Thus, RS codes match the Singleton bound, which implies that $\delta \leqslant \dfrac{n-k+1}{n}$.
Note that all the above properties do not depend on the choice of $\alpha_i' s$.

1.2 Decoding algortithms

Let $r = (r_1, r_2, \ldots, r_n) \in \mathbb{F}^n$ be the received message. We want to find an RS codeword c s.t. $\Delta(c, r) < (n-k+1)/2$ if it exists or say no. We will discuss the Welch Berlekamp algorithm. This algortihm is described in two steps.

Step 1 : Find a non-zero polynomial $Q(x, y) = A(x) + yB(x)$ s.t.

1. $\deg(A) < (n+k)/2 := D$
2. $\deg(B) < (n-k+2)/2 := d$
3. $Q(\alpha_i, r_i) = 0$ for all $1 \leq i \leq n$

Step 2 : Set $g(x) = -\dfrac{A(x)}{B(x)}$. If h is a polynomial of degree at most $k-1$ and the $\Delta(\text{Enc}(g), r) < (n-k+1)/2$ then output g else say no.

Let's first discuss how to do step 1. Observe that Q can be written as

$$Q(x, y) = A_0 + A_1 x + A_2 x^2 + \cdots + A_{D-1} x^{D-1} + B_0 y + B_1 xy + B_2 x^2 y + \cdots + B_{d-1} x^{d-1}$$

Observe that $D + d > n$. Now, $Q(\alpha_i, r_i) = 0$ for all i gives n equations in $D + d > n$ variables. Thus, it has a non-trivial solution which can be found using Gaussian Elimination. Then we check if $-A(x)/B(x)$ is a polynomial satisfying the above conditions and output accordingly. We will now argue the algorithm's correctness.

1.3 Correctness

Observe that if the algorithm outputs something, it is correct! This is true as we are checking for all the desired conditions in the final step. We thus want to argue that if there is a polynomial h satisfying the desired conditions, then the algorithm outputs it.

Claim 11.1. *If there exists a polynomial $h(x)$ with $\deg(h) \leqslant k-1$ such that $\Delta(\text{Enc}(h), r) < (n-k+1)/2$, then the algorithm outputs it.*

Proof. Let Q be a poly satisfying the constraints be some output of step 1. Consider

$$U(x) = Q(x, h(x)) = A(x) + h(x)B(x).$$

Observe that

1. $\deg(U) \leqslant \max(D-1, (k-1)+(d-1)) = \dfrac{n+k-2}{2}$ and
2. for any $1 \leq i \leq n$, $(h(\alpha_i) = r_i \implies U(\alpha_i) = 0)$.

Thus, the number of zeros of U is at least the number of agreements between r and $\text{Enc}(h)$. So, if the number of agreements of $\text{Enc}(h)$ and r is at least $\deg(U)$ then $U \equiv 0$. We know that $\Delta(\text{Enc}(h), r) < (n-k+1)/2$, and hence, the number of agreements between r and $\text{Enc}(h)$ is more than $(n+k-1)/2$, which is more than $\deg(U)$. Thus, $U \equiv 0$ and hence $g(x) = -\dfrac{A(x)}{B(x)} = h(x)$ is a polynomial of degree at most $k-1$ and will be output. □

1.4 Going beyond minimum distance

In the next class, we will look at decoding RS codes beyond the half distance limit. Unfortunately, in this scenario, uniqueness of such a codeword is not guaranteed. So, we need to define precisely as to what do we want the algorithm to do. One way is to ask the algorithm to output any code that satisfies the constraints. Other possibility is to ask the algorithm to output every codeword that satisfies the constraints. This, is called "List Decoding" (as opposed to Unique Decoding). We must ensure that the output list is of polynomial size. That is guaranteed by Johnson's theorem.

Johnson's Theorem :
Let $C \subseteq \mathbb{F}^n$ be a code with minimum distance Δ, then for all $r \in \mathbb{F}^n$, all the words in C within distance $n - \sqrt{n(n-\Delta)}$ is at most $poly(n)$.
Observe that for RS codes, $\Delta = n - k + 1$. Take $k \approx 0.01n$. We then get, $n - \sqrt{n(n-\Delta)} \approx n - \sqrt{nk} \approx 0.9n$.