25. Reed-Solomon Codes

25.1 Introduction

In the next few lectures, we study an algebraic code which has wide variety of applications in computer science. These are the Reed-Solomon codes. First we prove a bound on the maximum distance attainable by any $[n, k, d]_q$-code.

25.2 Singleton bound

The following bound, proved by Singleton, gives the distance achievable any $[n, k, d]_q$ code.

Theorem 25.1. *If $\mathcal{C}$ is an $[n, k, d]_q$ code, then $d \leq n - k + 1$.*

Proof. Let $\pi : [q]^n \to [q]^{k-1}$ be a projection of a vector $\mathbf{x}$ to its first $k-1$ positions. Since there are q^k codewords and the range of π has cardinality q^{k-1}, it must be the case that there exists two codewords $\mathbf{x}$ and $\mathbf{y}$ such that $\pi(\mathbf{x}) = \pi(\mathbf{y})$. This means that $\mathbf{x}$ and $\mathbf{y}$ agrees in the first $k-1$ positions , and hence $\Delta(\mathbf{x}, \mathbf{y}) \leq n - k + 1$. Since $d \leq \Delta(\mathbf{x}, \mathbf{y})$, the bound follows. □

The codes that achieve this bound are known as *Maximum Distance Separable* (MDS) codes. A natural question to ask here is whether there exists MDS codes. We will study Reed-Solomon (RS) codes and see that there are MDS.

25.3 Reed-Solomon Codes

Let $\mathbb{F}_q$ be a finite field and let n and k be such that $k \leq n \leq q$. We will now define the $[n, k, d]_q$ RS codes. Note that the alphabet size is greater that n. This is necessary for RS codes. It is also known (we will not prove it here) that if you want to construct MDS codes, the dependence of n on q is unavoidable.

Encoding

Choose $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}_q$ (we can do this because $q \geq n$). Let $\mathbf{m} = \mathbf{m}_0\mathbf{m}_1 \ldots \mathbf{m}_{k-1}$ be the message where $\mathbf{m}_i \in \mathbb{F}_q$. The message $\mathbf{m}$ therefore encodes a polynomial $P_{\mathbf{m}}[x] \in \mathbb{F}_q[x]$ as follows: $P_{\mathbf{m}} = \sum_{i=0}^{k-1} \mathbf{m}_i x^i$. Now, evaluate $P_{\mathbf{m}}$ at each of the points $\alpha_1, \ldots, \alpha_n$. The RS encoding of $\mathbf{m}$ is the string $(P_{\mathbf{m}}(\alpha_1), P_{\mathbf{m}}(\alpha_2), \ldots, P_{\mathbf{m}}(\alpha_n))$.

First, let us see that RS code is a linear code.

Theorem 25.2. *An $[n, k, d]_q$ Reed-Solomon code is a linear code.*

Proof. To prove this, we need to show that $\mathsf{RS}(\mathbf{m}^1) + \mathsf{RS}(\mathbf{m}^2)$ is a codeword, and that for every $\alpha \in \mathbb{F}_q$, $\alpha\mathsf{RS}(\mathbf{m})$ is also a codeword. First observe that $\mathsf{RS}(\mathbf{m}^1) + \mathsf{RS}(\mathbf{m}^2) = (P_{\mathbf{m}^1}(\alpha_1) + P_{\mathbf{m}^2}(\alpha_1), \ldots, P_{\mathbf{m}^1}(\alpha_n) + P_{\mathbf{m}^2}(\alpha_n))$. This corresponds to the encoding of the message $\mathbf{m} = (\mathbf{m}^1_1 + \mathbf{m}^2_2, \ldots, \mathbf{m}^1_k + \mathbf{m}^2_k)$. Also $\alpha\mathsf{RS}(\mathbf{m})$ corresponds to the encoding of the message $\mathbf{m} = (\alpha\mathbf{m}_1, \ldots, \alpha\mathbf{m}_k)$. □

How does the generator matrix for this linear code look? The codeword is i^{th} bit of the codeword is the evaluation of $P_{\mathbf{m}}$ at α_i. Therefore, the generator matrix is the Vandermonde matrix written as follows:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}.$$

Distance of the RS code

First let us note a few properties about polynomials over a finite field. From this we will derive the distance of the code.

Theorem 25.3. *Let $P(x)$ be a non-zero polynomial of degree d over the finite field $\mathbb{F}_q$. Then $P(x)$ has at most d roots.*

Proof. We prove this by induction on the degree of $P(x)$. The case when $d = 0$ is obvious then the polynomial $P(x)$ is a constant polynomial. Let $d > 0$, and let $\alpha \in \mathbb{F}_q$ be a root of $P(x)$. Then, we can write $P(x) = (x - \alpha)Q(x)$ where $Q(x)$ has degree at most $d - 1$. By the induction hypothesis $Q(x)$ has at most $d - 1$ roots in $\mathbb{F}_q$, and therefore $P(x)$ has at most d roots in $\mathbb{F}_q$. □

Now, we can prove that RS codes have distance $n - k + 1$.

Theorem 25.4. *The distance of an $[n, k, d]_q$ Reed-Solomon code is $n - k + 1$.*

Proof. Let $\mathsf{RS}(\mathbf{m})$ and $\mathsf{RS}(\mathbf{m}')$ be two codewords, where $\mathbf{m} = \mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_k$ and $\mathbf{m}' = \mathbf{m}'_1\mathbf{m}'_2 \ldots \mathbf{m}'_k$ are encoded as $(P_{\mathbf{m}}(\alpha_1), P_{\mathbf{m}}(\alpha_2), \ldots, P_{\mathbf{m}}(\alpha_n))$ and $(P_{\mathbf{m}'}(\alpha_1), P_{\mathbf{m}'}(\alpha_2), \ldots, P_{\mathbf{m}'}(\alpha_n))$ respectively. Consider $\mathsf{RS}(\mathbf{m}) - \mathsf{RS}(\mathbf{m}')$. This corresponds to the polynomial $P_{\mathbf{m}} - P_{\mathbf{m}'}$ over $\mathbb{F}_q$ and this is again of degree at most $k - 1$. From the theorem above, we know that it has at most $k - 1$ roots. Therefore, the codewords agree on at most $k - 1$ position, and has distance $d \geq n - k + 1$. From the Singleton bound, we know that for every $[n, k, d]_q$ code, $d \leq n - k + 1$. Therefore, Reed-Solomon code is an $[n, k, n - k + 1]_q$ linear code. □

Systematic view of the Reed-Solomon code

We had discussed earlier that every linear code can be thought of as sending the message bits followed by the parity check bits. In the case of Reed Solomon codes also, we can look

at it in this alternative way. Let $\mathbf{m} = \mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{k-1}$ be the message. The polynomial that we want is $P(x)$ such that $P(\alpha_i) = \mathbf{m}_i$ for $i \in \{0, 1, \ldots, k-1\}$. There is a unique degree $d-1$ polynomial that satisfies this condition and is given by the Langrangian interpolation formula.

$$P(x) = \sum_{j=0}^{k-1} \mathbf{m}_j \prod_{i \neq j} \frac{x - \alpha_i}{\alpha_j - \alpha_i}.$$

An example

Suppose we are working over the field $\mathbb{F}_3$ and we want to construct the $[3, 2, 2]_3$ Reed-Solomon code. The messages we want to encode are $(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)$. These correspond to the polynomials $0, 1, 2, X, X+1, X+2, 2X, 2X+1, 2X+2$ respectively. The corresponding codewords are obtained by evaluating these at $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 2$. Therefore, the actual codewords are $(0,0,0), (1,1,1), (2,2,2), (0,1,2), (1,2,0), (2,0,1), (0,2,1), (1,0,2), (2,1,0)$.