Chapter 5

Reed-Solomon codes and their decoding

Reed-Solomon codes were discovered in 1959 and have since then been one of the most important classes of error-correcting codes. The applications range from CDs and DVDs to satellite communications. In this chapter we will describe the Reed-Solomon codes and also give some decoding algorithms of these codes.

5.1 Basic definitions

Before introducing the codes we will first prove an upper bound on the minimum distance of any code.

Theorem 5.1.1. *(The Singleton bound) Let C be an (n, k) code with minimum distance d. Then*

$$d \leq n - k + 1$$

Proof. We give three different proofs of the theorem, in one of which we do not assume that the code is linear.

1) Choose the information vector to consist of $k - 1$ zeroes and one nonzero. Then the weight of the corresponding codeword is at most $(n - k) + 1$.

2) The rank of a parity check matrix H for the code is $n - k$ so $n - k + 1$ columns of H are linearly dependent. Therefore the minimum number of dependent columns (i.e. d) must be smaller than or equal to $n - k + 1$.

3) If we delete $d - 1$ fixed positions from all the q^k codewords, they are still different since each pair differ in at least d positions. There are q^{n-d+1} vectors with the remaining positions and thus $k \leq n - d + 1$ and the result follows. □

Definition 5.1.1. *(Reed-Solomon codes)*
Let $x_1, \ldots, x_n$ be different elements of a finite field $\mathbb{F}_q$. For $k \leq n$ consider the set $\mathbb{P}_k$ of polynomials in $\mathbb{F}_q[x]$ of degree less than k. A Reed-Solomon *code consists of the codewords*

$$(f(x_1),\, f(x_2), \ldots,\, f(x_n)) \text{ where } f \in \mathbb{P}_k$$

It is clear that the length of the code is $n \leq q$. The code is linear since if

$$c_1 = (f_1(x_1), \ldots, f_1(x_n)) \quad \text{and} \quad c_2 = (f_2(x_1), \ldots, f_2(x_n)),$$

then

$$ac_1 + bc_2 = (g(x_1), \ldots, g(x_n))$$

where $a, b \in \mathbb{F}_q$ and $g(x) = af_1(x) + bf_2(x)$.

The polynomials in $\mathbb{P}_k$ form a vector space over $\mathbb{F}_q$ of dimension k since there are k coefficients. We now invoke a fundamental theorem of algebra (Theorem 2.2.1) twice: Two distinct polynomials cannot generate the same codeword since the difference would be a polynomial of degree $< k$ and it cannot have n zeroes, so the dimension of the code is k. A codeword has weight at least $n - k + 1$ since a polynomial of degree $< k$ can have at most $k - 1$ zeroes. Combining this with Theorem 5.1.1 we get

Theorem 5.1.2. *The minimum distance of an (n, k) Reed-Solomon code is $n - k + 1$.*

In many applications one takes $x_i = \alpha^{i-1}$, $i = 1, 2, \ldots, q - 1$ where α is a primitive element of $\mathbb{F}_q$, so in this case we have $x_i^n = 1$, $i = 1, \ldots, n$, and $n = q - 1$.

From the definition of the codes it can be seen that one way of encoding these codes is to take k information symbols $i_0, i_1, \ldots, i_{k-1}$ and encode them as

$$(i(x_1), i(x_2), \ldots, i(x_n))$$

where $i(x) = i_{k-1}x^{k-1} + \cdots + i_1 x + i_0$.

This is a non-systematic encoding; we will describe a systematic encoding in a problem in the next chapter.

Since for Reed-Solomon Codes we must have $n \leq q$ there are no interesting binary codes. Codes over $\mathbb{F}_q$ where q is a prime make easier examples and in particular the field $\mathbb{F}_{11}$ is useful for decimal codes, since there is no field with ten elements. However in most practical cases we have $q = 2^m$.

Example 5.1.1. Reed-Solomon codes over $\mathbb{F}_{11}$

Since 2 is a primitive element of $\mathbb{F}_{11}$ we can take $x_i = 2^{i-1} \bmod 11, i = 1, 2, \ldots, 10$ with $k = 5$ and $i(x) = i_4x^4 + \cdots + i_1 x + i_0$; we get as the corresponding codeword as

$$(i(1), i(2), i(4), i(8), i(5), i(10), i(9), i(7), i(3), i(6))$$

So

(1, 0, 0, 0, 0)	is encoded into	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
(0, 1, 0, 0, 0)	is encoded into	(1, 2, 4, 8, 5, 10, 9, 7, 3, 6)
(0, 0, 1, 0, 0)	is encoded into	(1, 4, 5, 9, 3, 1, 4, 5, 9, 3)
(0, 0, 0, 1, 0)	is encoded into	(1, 8, 9, 6, 4, 10, 3, 2, 5, 7)
(0, 0, 0, 0, 1)	is encoded into	(1, 5, 3, 4, 9, 1, 5, 3, 4, 9)

and these five codewords can be used as the rows of a generator matrix of the code, which is a $(10, 5, 6)$ code over $\mathbb{F}_{11}$.

5.2 Decoding Reed-Solomon Codes

In this section we describe the first of three minimum distance decoding algorithms for Reed-Solomon codes (in Chapter 12 we present an algorithm for correcting more errors). We will first present the idea and give a formal algorithm later.

Let $r = c + e$ be a received word, and assume that $w(e) \leq t = \lfloor \frac{n-k}{2} \rfloor$. The idea is to determine a bivariate polynomial

$$Q(x, y) = Q_0(x) + yQ_1(x) \in \mathbb{F}_q[x, y] \backslash \{0\}$$

such that

1. $Q(x_i, r_i) = 0, i = 1, \ldots, n.$
2. $\deg(Q_0) \leq n - 1 - t.$
3. $\deg(Q_1) \leq n - 1 - t - (k - 1).$

The polynomial $Q(x, y)$ is called an interpolating polynomial for the received word. We first prove:

Theorem 5.2.1. *There is at least one nonzero polynomial $Q(x, y)$ which satisfies conditions $1 - 3$.*

Proof. The condition 1 gives n homogeneous linear equations in the coefficients and there are $n - 1 - t + 1 + n - 1 - t - (k - 1) + 1 \geq n + 1$ possible coefficients, so indeed the system has a nonzero solution. □

We also have

Theorem 5.2.2. *If the transmitted word is generated by $g(x)$ and the number of errors is less than $\frac{d}{2}$, then $g(x) = \frac{-Q_0(x)}{Q_1(x)}$.*

Proof. $c = (g(x_1), \ldots, g(x_n))$ and $r = c + e$ with $w(e) \leq t$. The polynomial $Q(x, y)$ satisfies $Q(x_i, g(x_i) + e_i) = 0$ and since $e_i = 0$ for at least $n - t$ is, we see that the univariate polynomial $Q(x, g(x))$ has at least $n - t$ zeroes, namely the x_is where $g(x_i) = r_i$. But $Q(x, g(x))$ has degree at most $n - t - 1$, so $Q(x, g(x)) = 0$ and therefore $Q_0(x) + g(x)Q_1(x) = 0$ and hence $g(x) = -\frac{Q_0(x)}{Q_1(x)}$. □

The maximal degrees of the components of Q will be used frequently in the following, so we define

$$l_0 = n - 1 - t \quad \text{and} \quad l_1 = n - 1 - t - (k-1)$$

Note that since $Q(x, y) = Q_1(x)\left(y + \frac{Q_0(x)}{Q_1(x)}\right) = Q_1(x)(y - g(x))$ the $x_i's$ where the errors occurred are among the zeroes of $Q_1(x)$, therefore the polynomial $Q_1(x)$ is called an *error locator polynomial*.

The algorithm now can be presented as follows:

Algorithm 5.2.1.

Input: *A received word* $r = (r_1, r_2, \ldots, r_n)$

1. Solve the system of linear equations

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{l_0} & r_1 & r_1x_1 & \ldots & r_1x_1^{l_1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{l_0} & r_2 & r_2x_2 & \ldots & r_2x_2^{l_1} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{l_0} & r_n & r_nx_n & \ldots & r_nx_n^{l_1} \end{bmatrix} \begin{bmatrix} Q_{0,0} \\ Q_{0,1} \\ Q_{0,2} \\ \vdots \\ Q_{0,l_0} \\ Q_{1,0} \\ Q_{1,1} \\ \vdots \\ Q_{1,l_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.1}$$

2. Put

$$Q_0(x) = \sum_{j=0}^{l_0} Q_{0,j}x^j, \qquad Q_1(x) = \sum_{j=1}^{l_1} Q_{1,j}x^j, \qquad g(x) = -\frac{Q_0(x)}{Q_1(x)}$$

3. If $g(x) \in \mathbb{F}_q[x]$

output: $(g(x_1), g(x_2), \ldots, g(x_n))$

else

output: failure

Notice in the system above that each row of the matrix corresponds to a pair (x_i, r_i). We have already seen that if the number of errors is smaller than half the minimum distance, then the output of the algorithm is the sent word.

Example 5.2.1. Decoding the (10, 5, 6) Reed-Solomon code over $\mathbb{F}_{11}$

We treat the code from Example 5.1.1 and suppose we receive $r = (5, 9, 0, 9, 0, 1, 0, 7, 5)$. We have $l_0 = 7$ and $l_1 = 3$ and therefore we get 10 equations with 11 unknowns. The matrix becomes

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 5 & 5 & 5 & 5 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 & 9 & 7 & 3 & 6 \\ 1 & 4 & 5 & 9 & 3 & 1 & 4 & 5 & 0 & 0 & 0 & 0 \\ 1 & 8 & 9 & 6 & 4 & 10 & 3 & 2 & 9 & 6 & 4 & 10 \\ 1 & 5 & 3 & 4 & 9 & 1 & 5 & 3 & 0 & 0 & 0 & 0 \\ 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 & 1 & 10 \\ 1 & 9 & 4 & 3 & 5 & 1 & 9 & 4 & 0 & 0 & 0 & 0 \\ 1 & 7 & 5 & 2 & 3 & 10 & 4 & 6 & 7 & 5 & 2 & 3 \\ 1 & 3 & 9 & 5 & 4 & 1 & 3 & 9 & 0 & 0 & 0 & 0 \\ 1 & 6 & 3 & 7 & 9 & 10 & 5 & 8 & 5 & 8 & 4 & 2 \end{bmatrix}$$

The system has as a solution

$(4, 1, 2, 2, 2, 9, 1, 0, 7, 3, 10, 0)$ corresponding to $Q_0(x) = x^6 + 9x^5 + 2x^4 + 2x^3 + 2x^2 + x + 4$ and $Q_1(x) = 10x^2 + 3x + 7$. We then get $g(x) = x^4 + x^3 + x^2 + x + 1$ corresponding to the codeword $c = (5, 9, 0, 6, 0, 1, 0, 7, 0, 4)$ so we have corrected two errors in positions corresponding to $2^3 (= 8)$ and $2^9 (= 6)$ and one sees that indeed 8 and 6 are the zeroes of $Q_1(x)$.

5.3 Vandermonde matrices

In this section we give some results for matrices with a special structure. As we will demonstrate these matrices play a significant role in the following, in particular we find parity check matrices for Reed-Solomon codes.

Lemma 5.3.1. *Let $\beta \in \mathbb{F}_q$ be an element of order n, i.e. $\beta^n = 1$ and $\beta^i \neq 1$ for $0 < i < n$ and let $x_j = \beta^{j-1}$, $j = 1, 2, \ldots, n$. Let*

$$A = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \ldots & \vdots \\ x_1^a & x_2^a & \ldots & x_n^a \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ldots & \vdots \\ x_1^s & x_2^s & \ldots & x_n^s \end{bmatrix}$$

where

$$s + a + 1 \leq n$$

then

$$BA^T = 0$$

Before we prove the claim we note that it also follows that $AB^T = 0$.