A LITTLE BIT ABOUT REED-SOLOMON CODES

Contents

1. Introduction

This document will introduce you to Reed-Solomon codes. We will do this different from the book, so please read this and not section 18.9 of the book. Also, please reload before reading.

2. Construction of the codes

Let $\mathbf{F}$ be a finite field with q elements. Recall that $q = p^f$ for some integer $f \geq 1$ and prime number p.

Let $1 \leq k \leq n \leq q$. Choose pairwise distinct elements $a_1, \ldots, a_n \in \mathbf{F}$.

Given a polynomial $m = m_1 + m_2 x + \ldots + m_k x^{k-1}$ of degree $\leq k-1$ we get a codeword

$$c(m) = (c_1, \ldots, c_n) = (m(a_1), m(a_2), \ldots, m(a_n))$$

by taking the values of our polynomial at our chosen points. Observe that if m, m' are two polynomials of degree $\leq k-1$ and if $\lambda, \lambda' \in \mathbf{F}$ are two scalars, then we have

$$c(\lambda m + \lambda' m') = \lambda c(m) + \lambda' c(m')$$

Thus we see that the set of all codewords obtained in this manner is a subspace of $\mathbf{F}^n$. In other words, this set defines a linear code and this is the Reed-Solomon code.

Definition 2.1. Let q and $1 \leq k \leq n \leq q$ be as above. Let $a_1, \ldots, a_n \in \mathbf{F}$ be pairwise distinct. The q-ary linear **Reed-Solomon** $[n, k]$-code is the set of all codewords $c(m)$ constructed above.

It is clear that the length of the codewords in a Reed-Solomon code is n. To make sure our notation is consistent we will now check that the dimension is k.

3. The dimension of a Reed-Solomon code

Let $C \subset \mathbf{F}^n$ be a Reed-Solomon code constructed using a field with q elements and $1 \leq k \leq n \leq q$ as above. We claim that C has dimension k. To see this it suffices (and in fact it is also necessary) to show that C has q^k elements (see earlier lecture). Now for every polynomial m of degree $\leq k-1$ we get a codeword

$$c(m) = (m(a_1), m(a_2), \ldots, m(a_n)) \in C$$

The number of possible polynomials $m = m_1 + m_2 x + \ldots + m_k x^{k-1}$ is q^k because we have q choices for each coefficient m_i of m. Thus it suffices to show: if m, m' are two distinct polynomials of degree $\leq k-1$, then $c(m) \neq c(m')$. From what was said above we have

$$c(m) = c(m') \Leftrightarrow c(m - m') = 0$$

Thus this happens if and only if the polynomial $m - m'$ is zero in all the points a_i, $i = 1, \ldots, n$. Thus all we have to do is show that a polynomial of degree $k-1$ cannot vanish in n points. This is true by the following lemma which you should accept as true.

Lemma 3.1. *Let $\mathbf{F}$ be any field. A polynomial f of degree d can have at most d distinct zeros in $\mathbf{F}$.*

Proof. Let $a \in \mathbf{F}$. We claim that $f(a) = 0$ if and only if the polynomial f is divisible by the linear polynomial $x - a$. Namely, by division with remainder we can write

$$f(x) = q(x)(x - a) + r(x)$$

in $\mathbf{F}[x]$ where $r(x)$ has degree < 1. This means that $r(x) = r$ is a constant polynomial. Substituting $x = a$ in both sides we see that $r = 0$ if and only if $f(a) = 0$. This proves the claim.

Now if $f(a_1) = 0, \ldots, f(a_t) = 0$ for pairwise distinct elements $a_1, \ldots, a_t \in \mathbf{F}$, then we can first write

$$f(x) = (x - a_1)g(x)$$

by the claim above. Note that the degree of g is exactly one less than the degree of f. Substituting $x = a_i$ for $i > 1$ we conclude that $g(a_i) = 0$ for $i = 2, \ldots, t$. Here we use that a_i does not equal a_1 for $i > 1$ because we are dividing by $a_i - a_1$ to see this. Continuing in this fashion (or by using induction) we find that

$$f(x) = (x - a_1) \ldots (x - a_t)h(x)$$

for some polynomial $h(x)$ of degree equal to the degree of $f(x)$ minus t. Thus certainly the degree of f has to be at least t. □

4. Generating matrix

This section is just showing you how to think about Reed-Solomon codes in terms of generating matrices and to show you some examples.

Suppose that $p = 3$ and we take $k = 2$ and $n = 3$. As our points a_1, a_2, a_3 we take $0, 1, 2$. The codeword associated to the constant polynomial $m = 1$ is

$$c(1) = (1, 1, 1)$$

The codeword associated to $m = x$ is

$$c(x) = (0, 1, 2)$$

Thus we see that

$$G = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

is a generator matrix for our Reed-Solomon code.

Suppose that $p = 7$ and we take $k = 4$ and $n = 7$. Then we take our points $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ to be $0, 1, 2, 3, 4, 5, 6$. As before we can use the polynomials $1, x, x^2, x^3$ to get the rows of the generator matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 4 & 2 & 2 & 4 & 1 \\ 0 & 1 & 1 & 6 & 1 & 6 & 6 \end{pmatrix}$$

5. Minimum distance

In this section we compute the minimum distance

Lemma 5.1. *Let $C \subset \mathbf{F}^n$ be a Reed-Solomon code constructed using a field with q elements and $1 \leq k \leq n \leq q$ as above. The minimum distance of C is $d = n - k + 1$.*

Proof. Recall that the minimum distance of a linear code C is the minimal Hamming weight of a nonzero element of C. A nonzero polynomial of degree $k-1$ has at most $k-1$ distinct zeros (see Lemma 3.1). Hence a nonzero codeword has at least $n - (k-1)$ nonzero entries. In this way we see that the minimum distance is $\geq n - k + 1$.

Moreover, the minimum distance is actually equal to this. Namely, we can take

$$m = (x - a_1)(x - a_2) \ldots (x - a_{k-1})$$

This is a polynomial of degree $k-1$. The codeword associated to m is

$$c(m) = (0, \ldots, 0, m(a_k), m(a_{k+1}), \ldots, m(a_n))$$

which starts with exactly $k-1$ entries equal to 0. □

This is pretty amazing, because this means that Reed-Solomon codes are *maximum distance separable* codes (largest possible value of d for given n and M).

Corollary 5.2. *Let $C \subset \mathbf{F}^n$ be a Reed-Solomon code constructed using a field with q elements and $1 \leq k \leq n \leq q$ as above. Then C is a MDS code, i.e., it achieves equality in the Singleton bound.*

Proof. The singleton bound is

$$M \leq q^{n-d+1}$$

where M is the number of codewords. For our code we have $M = q^k$ and $d = n - k + 1$. Thus we have equality. □

6. Adding the point at infinity

It turns out you can use the idea explained above to get codes with $1 \leq k \leq n = q + 1$. Namely, say $\mathbf{F} = \{a_1, \ldots, a_q\}$ is a listing of all the elements. Let $m = m_1 + m_2 x + \ldots + m_k x^{k-1}$ be a polynomial of degree $\leq k-1$. We are going to think of the leading coefficient m_{k-1} as the value of m at the point $x = \infty$. Thus given m we consider the "extended" codeword

$$c(m) = (m(a_1), m(a_2), \ldots, m(a_q), m_{k-1})$$

As before you can show that this gives a q-ary linear $[q+1, k]$-code. The minimum distance of this code is $n - k + 1 = (q+1) - k + 1 = 2 + q - k$. To see this argue as in the proof of Lemma 5.1 but now use that it $m_{k-1} = 0$, then of course the polynomial has degree $< k - 1$. Hence these are also MDS codes.

Example: say $q = 3$ and $k = 2$. As our points a_1, a_2, a_3 we take $0, 1, 2$. The codeword associated to the constant polynomial $m = 1$ is $c(1) = (1, 1, 1, 0)$ The codeword associated to $m = x$ is $c(x) = (0, 1, 2, 1)$. Thus we see that

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix}$$

is a generator matrix for our "extended" Reed-Solomon code. This is a 3-ary $[4, 2, 3]$-code.

Example: suppose that $p = 7$ and we take $k = 4$. Then we take our points $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ to be $0, 1, 2, 3, 4, 5, 6$. As before we can use the polynomials $1, x, x^2, x^3$ to get the rows of the generator matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 0 & 1 & 4 & 2 & 2 & 4 & 1 & 0 \\ 0 & 1 & 1 & 6 & 1 & 6 & 6 & 1 \end{pmatrix}$$

This is a 7-ary $[8, 4, 5]$-code.

7. Application of Reed-Solomon codes

Suppose that $\mathbf{F} = GF(2^8) = \mathbf{F}_{2^8}$ is the field with 256 elements. Take either

(1) $n = 255$ (this is what they do in the book on page 441, but please don't read there), or
(2) $n = 256$ (see above), or
(3) $n = 257$ (see Section 6).

Say we want to correct 16 errors. Then we take $d = 33$ and hence $k = n + 1 - 33 = 223$ (in the first case — this is the case used in practice where one uses Reed-Solomon codes presented as cyclic codes as in the book — don't read this), or $k = 224$ (in the second case), or $k = 225$ (in the third case).

In the first case for example, we transmit a codeword as a string of bits of length $8 \times 255 = 2040$. The advantage of this is that if all errors occur in a substring consecutive bits of length 121, then this can affect at most 16 of the letters of the codeword and hence this can be error corrected. In certain applications it is more likely that all the errors occur in a burst (for example if a cd is scratched or when a radio transmission is interrupted by some short term physical event).

8. Drawbacks of Reed-Solomon codes

First drawback: The length of the codewords n is bounded by q.

Second drawback: It is not trivial to do the error correcting, although there are fairly good algorithms.

9. Algebraic Geometry Goppa codes

You can use the analogue of the idea for the Reed-Solomon codes given above for more interesting rings of functions. Instead of having one variable x you'll have two variables x, y and these will satisfy some algebraic relation; this relation will define an algebraic curve X over the finite field $\mathbf{F}$. Then we will have a collection of polynomials m in x, y which we will evaluate at some points $a_1, \ldots, a_n \in C$. Just as before some estimates on the number of zeros will give a bound on the minimum distance of the code you obtain.

Using this (and a lot more theory), Tsfasman, Vladut, and Zink found sequences of codes for fixed square $q \geq 49$ and n/d tending to a fixed ratio in $(0, 1 - 1/q)$ which beat the asymptotic Gilbert-Varshamov bound! (As far as I know it is still open if you can do the same with $q = 2$.)

10. Example of an Algebraic Geometry Goppa code

This example is taken from lecture notes by Rachel Pries with slight modifications.

Let's work over $\mathbf{F}_4 = \{0, 1, \omega, \omega + 1\}$ where $\omega^2 = \omega + 1$ as in the book on page 362. Consider the elliptic curve

$$E : y^2 + y = x^3$$

which has 8 points in the affine plane, namely

$$\begin{aligned} a_1 &= (0, 0), \\ a_2 &= (0, 1), \\ a_3 &= (1, \omega), \\ a_4 &= (1, \omega + 1), \\ a_5 &= (\omega, \omega), \\ a_6 &= (\omega, \omega + 1), \\ a_7 &= (\omega + 1, \omega), \\ a_8 &= (\omega + 1, \omega + 1) \end{aligned}$$

As you know there is also the point at ∞; we will come back to this later. Let's take the points $a_1, \ldots, a_8$ and the polynomial functions $1, x, y$. Exactly as before you get the generating matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \omega & \omega & \omega + 1 & \omega + 1 \\ 0 & 1 & \omega & \omega + 1 & \omega & \omega + 1 & \omega & \omega + 1 \end{pmatrix}$$

Then you can show that this is a 4-ary linear $[8, 3, 5]$-code. For example the polynomial $m = \omega + y$ gives the codeword

$$c(m) = (\omega, \omega + 1, 0, 1, 0, 1, 0, 1)$$

which has Hamming weight 5 hence the minimum distance cannot be greater than 5. You can check all polynomials one-by-one and show that none of them vanish in more than 3 of the points a_i. But actually, this is just the statement that a line, i.e., a subset of $\mathbf{F}^2$ defined by an equation of the form

$$\lambda + \mu x + \nu y = 0, \text{ for some } \lambda, \mu, \nu \in \mathbf{F} \text{ not all zero,}$$

doesn't intersect the elliptic curve E in more than 3 points exactly as we've seen in our deliberations about points on elliptic curves earlier in the course.

Comments on the example. It turns out that $d = 5$ which is best possible for 4-ary linear $[8, 3]$ codes (you can find tables on the site www.codetables.de if you are interested). On the other hand, a $[8, 3, 5]$-code is not an MDS code (in other words, 4-ary linear MDS codes with $n = 8$ and $k = 3$ do not exist).

Another comment is that if you "add the point at infinity" to extend the code above as in Section 6 (to really understand what this means you'll need more theory; to do this in practice you can just guess an extra column to add to the matrix), then you'll obtain a 4-ary $[9, 3, 6]$-code (here minimum distance 6 is also best possible for for 4-ary linear $[9, 3]$ codes).

It turns out that there do not exist linear 4-ary $[10, 3, 7]$-codes. So it shouldn't be possible to find an elliptic curve over $\mathbf{F}_4$ with 10 points. And indeed Hasse's theorem on page 354 of the book tells us this isn't possible.