Sub class of cyclic codes, frequently used.

Recall about finite fields theory

Lemma

In any finite field $\mathbb{F}_q$ $\exists \alpha \in \mathbb{F}_q \mid \forall \beta \in \mathbb{F}_q \setminus \{0\}$ $\beta = \alpha^i$ $i \in \mathbb{N}$

α is called primitive element

Note that taking higher powers of α makes you go back to the same elements

$\alpha^{q-1} = \alpha^0 = 1$, $\alpha^q = \alpha^1$ etc.

this implies that for every non zero element α^K we have $(\alpha^K)^{q-1} = (\alpha^{q-1})^K = 1^K = 1$

hence, over $\mathbb{F}_q$, the poly $X^{q-1}-1$ can be easily factorised

$$X^{q-1}-1 = \prod_{i=0}^{q-2} (X-\alpha^i)$$

A Reed-Solomon code is a cyclic code of length $q-1$ over the field $\mathbb{F}_q$ with a generator polynomial

$$g(x) = \prod_{j=0}^{2t-1} (X-\alpha^j)$$ (α: primitive element of $\mathbb{F}_q$)

theorem the parameters of a Reed-Solomon code are $[q-1, q-2t-1, 2t+1]$

proof: the first parameter is by definition

the second one is $q-1-2t$ because $\deg(g(x)) = 2t$

we have to prove that all code words have $w > 2t$

suppose $e(x) = \sum_{l=1}^{v} e_{i_l} X^{i_l}$ is a poly. of non zero weight $v < 2t+1$, this can never be a code word

because otherwise $e(x) = g(x)m(x)$ and $e(\alpha^j) = g(\alpha^j)m(\alpha^j) = 0$ for $j \leq 2t-1$

so we would have equations

$$e_{i_1}(\alpha^0)^{i} + \cdots + e_{i_v}(\alpha^0)^{i_v} = 0$$
$$\vdots \qquad\qquad \vdots$$
$$e_{i_1}(\alpha^{v-1})^{i} + \cdots + e_{i_v}(\alpha^{v-1})^{i_v} = 0$$

these can be solved uniquely as long as the following det $\neq 0$

$$\det \begin{pmatrix} (\alpha^0)^{i_1} & \cdots & (\alpha^0)^{i_v} \\ \vdots & & \\ (\alpha^{v-1})^{i_1} & \cdots & (\alpha^{v-1})^{i_v} \end{pmatrix} = \prod_{k<\lambda} (\alpha^{i_k} - \alpha^{i_\lambda})$$

← Vandermonde det. formula

Because $\alpha^{i_k} \neq \alpha^{i_\lambda}$ whenever $i_k \neq i_\lambda$ and both exponents are smaller than $q-1$ the det is not zero and the only possible solution is $\bar{0}$.

So a poly with $w < 2t+1$ can never be a code word.

there is however a codeword with $w = 2t+1$: $g(x) = g_0 + g_1 x \cdots + g_{2t} x^{2t}$

theorem An $[n,k,d]$ Reed-Solomon code is maximal distance separable, i.e. it has the greatest possible minimal distance of all possible codes with the same n and K

proof: for a general linear $[n,k]$-code $d \leq n-k+1$

Since the parity check matrix has $n-K$ rows it has rank at most $n-K$.

So there's a linear combination of $n-k+1$ columns of H that gives us 0.

This linear combination can be written as $Hx^t=0$ for a certain vector x with weight $n-k+1$.

This implies that x is a code word and hence $d \leq n-K+1$.

For a RS-code $K=q-2t$ and $d=2t+1=(q-1)-(q-2t-1)+1=n-K+1$ so it has the largest minimal distance possible

This does not imply they are the best codes that exist.

there are better combination of n and K that don't give RS-codes

One of the disadvantages of RS-codes is that the size of the alphabet limits the size of the code

examples

1) $q=5$ $t=1$ a possible choice for α is 2, the pow. of 2 mod 5 are $1=2^0, 2=2^1, 4=2^2, 3=2^3$ (then they repeat)
this means $g(x)=(x-1)(x-2)=x^2+2x+2$ and $RS(5,1)$ is a linear $[4,2,3]$ code over $\mathbb{F}_5$

2) $q=7$ $t=2$ possible choice $\alpha=3$

$1=3^0, 3=3^1, 2=3^2, 6=3^3, 4=3^4, 5=3^5$

$g(x)=(x-1)(x-3)(x-2)(x-6)$ $RS(7,2)$ is a linear $[6,2,5]$ code over $\mathbb{F}_7$

Syndromes for RS codes

the gen. pol. has been chosen so that it has zeroes $1, \alpha, \alpha^2 \dots \alpha^{2t-1}$, α primitive element of $\mathbb{F}_q$

This implies that if $w(x)$ is the recieved poly. coming from a message $m(x)$ and an err. poly. $e(x)$ we get:

$$w(\alpha^i) = m(\alpha^i)g(\alpha^i) + e(\alpha^i) = e(\alpha^i), \quad i = 0, 2, \dots, 2t-1$$

for $j = 0, 1, \dots, 2t-1$ we define the j-th syndrome for the received poly. $r(x)$ as:

$$S_j = r(\alpha^j) = \sum_{l=1}^{n} e_l (\alpha^j)^l$$

suppose that a total of v errors occurred, located at positions $i_1, \dots, i_v$, $v \leq t$

$$e(x) = e_{i_1} x^{i_1} + \dots e_{i_v} x^{i_v}$$

then the powers of x define the error locations and the coefficients their magnitude.

we can now write

$$S_0 = e_{i_1}(\alpha^{i_1})^0 + e_{i_2}(\alpha^{i_2})^0 + \dots + e_{i_v}(\alpha^{i_v})^0$$

$$\vdots \qquad\qquad \vdots$$

$$S_{2t-1} = e_{i_1}(\alpha^{i_1})^{2t-1} + e_{i_2}(\alpha^{i_2})^{2t-1} + \dots + e_{i_v}(\alpha^{i_v})^{2t-1}$$

we need to solve for a^{i_l} and e_{i_l}

Once they have been found, we can take logarithms (base α) in $\mathbb{F}_q$:

$$\log_\alpha X = j \iff \alpha^j = X \quad \wedge \quad 1 \leq j \leq q-1$$

to find the error location numbers i_l, and correct the symbols at these position subtracting the magnitudes

the following is the same as BCH codes

⋮

~ · — · ~ · ~

<u>example</u> single error correction

if a single err. occurred, $S_0 = Y_0$

$S_1 = Y_1 X_1$

hence $X_1 = S_1/S_0$ and $Y_1 = S_0$ if $S_1/S_0 = \alpha^i$ the error is at location i and its value is S_0.

Finding errors and roots

- Syndrome polynomial -> $S(x) = \sum_{l=0}^{2t-1} S_l x^l$
- error location polynomial -> $\Lambda(x) = \prod_{l=1}^{v} (1 - X_l x)$
- error evaluator polynomial -> $\Omega(x) = \sum_{k=1}^{v} y_k \prod_{\substack{l=1 \\ l \neq k}}^{v} (1 - X_l x)$

to avoid confusion we can use $X_l = \alpha^{i_l}$ and $y_l = e_{i_l}$

$$S_1 = y_1 X_1 + y_2 X_2 + \cdots + y_v X_v$$
$$S_2 = y_1 X_1^2 \cdots$$
$$\vdots$$
$$S_{2t} = y_1 X_1^{2t} \cdots + y_v X_v^{2t}$$

where the S_i, $X_l = \alpha^{i_l}$, $y_l = e_{i_l}$ are defined as the previous section

Note that by definition the roots of the error-locator poly. are directly related to the error location variables

$\Lambda(x) = 0 \iff X = \alpha^{-i_l}$ for some error location i_l

The polynomial itself however is unknown, and must be determined (hopefully efficiently) for the syndrome polynomial.
After this we can calculate the roots of $\Lambda(x)$ to find the error locations.

Finding root of poly. is hard (in $\mathbb{R}$ no general sol. deg. ≥ 5) but finite fields have the advantage that we can find the roots by exhaustive search (brute force) over $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$.
This search is called Chien search
After we have found the roots of $\Lambda(x)$ we can determine the sizes of the errors using the error evaluator polynomial.

We will first recall formal derivations if $\mathbb{F}_q$

for a poly. $a(x) = \sum_{i=0}^{s} a_i x^i \to a'(x) = \sum_{i=1}^{s} i a_i x^{i-1}$

we take i modulo the characteristics of $\mathbb{F}_q$
This derivatives hold the same properties as the normal ones.

we can now calculate that

$$\Lambda'(x) = \sum_k -\alpha^{i_k} \prod_{l \neq k} (1 - \alpha^{i_l} x)$$

so by the definition of $\Omega(x)$

$$\Lambda'(\alpha^{-i_k}) = -\alpha^{i_k} \prod_{i \neq k} (1 - \alpha^{i_i - i_k}) = \frac{-\alpha^{i_k}}{e_{i_k}} \Omega(\alpha^{-i_k})$$

this gives the expression for the error magnitudes

$$e_{i_k} = \frac{-\alpha^{i_k} \Omega(\alpha^{-i_k})}{\Lambda'(\alpha^{-i_k})}$$

Problem Given the syndrome polynomial $S(x)$ determine the error locator polynomial $\Lambda(x)$ and the error evaluator polynomial $\Omega(x)$

⇓

Berlekamps algorithm

consider the ring $\mathbb{F}_q[x]/(x^{2t})$. The polynomial $1-\alpha^j X$ is not divisible by X and hence invertible

Its inverse is computed by chopping off the Taylor expansion at $x^{2t}\equiv 0$

$$\frac{1}{1-\alpha^j X} \equiv \sum_{l=0}^{2t-1} (\alpha^j X)^l \mod x^{2t}$$

using this identity we get

$$S(x) = \sum_{l=0}^{2t-1} S_l X^l = \sum_{l=0}^{2t-1} \sum_{k=1}^{v} e_{i_k} (\alpha^{i_k} X)^l =$$

$$= \sum_{k=1}^{v} e_{i_k} \left(\sum_{l=0}^{2t-1} (\alpha^{i_k} X)^l \right) \equiv \sum_{k=1}^{v} \frac{e_{i_k}}{1-\alpha^{i_k} X} \mod x^{2t}$$

we can relate $\Lambda(x), S(x)$ and $\Omega(x)$ in the following way

$$\Omega(x) = \Lambda(x) \sum_{k=1}^{v} \frac{e_{i_k}}{1-\alpha^{i_k} X} \equiv \Lambda(x) S(x) \mod x^{2t}$$

to find $\Omega(x)$ and $\Lambda(x)$ out of $S(x)$ we will use the algorithm of Euclid.

Suppose that $a(x)$, $b(x)$ polynomials of $\mathbb{F}_q$ then the algorithm of Euclid supplies us with series $r_i(x)$, $s_i(x)$ and $t_i(x)$ $i \in \{1 \dots k+1\}$ such that:

$$s_i(x)a(x) + t_i(x)b(x) = r_i(x) \quad \text{and} \quad \deg t_i(x) + \deg r_{i-1}(x) = \deg a(x)$$

with $r_0(x) = a(x)$ $r_k(x) = \gcd(a(x), b(x))$ and $r_{k+1}(x) = 0$

theorem Suppose $t(x)$ and $r(x)$ nonzero polynomials over $\mathbb{F}_q$ satisfying the following conditions:

1) $\gcd(t(x), r(x)) = 1$
2) $\deg t(x) + \deg r(x) < \deg a(x)$
3) $t(x)b(x) \equiv r(x) \mod(a(x))$

Then there exist an index $h \in \mathbb{N}$ and a constant $c \in \mathbb{F}_q$ such that

$$t(x) = c\,t_h(x) \quad \text{and} \quad r(x) = c\,r_h(x)$$

from Euclid's algorithm

proof: observe that $\deg r_i(x)$ strictly decreases when i increases.

by condition 2 we have that $\deg r(x) < \deg a(x)$ hence there exist an index h so that

$$\deg r_h(x) \leq \deg r(x) \leq \deg r_{h-1}(x)$$

from condition 3 we have that there is an $S(x) \in \mathbb{F}_q[x]$ such that

① $S(x)a(x) + t(x)b(x) = r(x)$

while from Euclides algorithm we have that

② $S_h(x)a(x) + t_h(x)b(x) = r_h(x)$

Multiplying ① by t_h and ② by $t(x)$ and subtracting the results we obtain

③ $(t(x)S_h(x) - t_h(x)S(x))a(x) = t(x)r_h(x) - t_h(x)r(x)$

now we use the definition of h to get a bound on the degrees, by condition 2 we have that

$$\deg t(x)r_h(x) = \deg t(x) + \deg r_h(x) \leq \deg t(x) + \deg r(x) < \deg(a)$$

and bcz of Euclid's algorithm:

$$\deg t_h(x)r(x) = \deg t_h(x) + \deg r(x) < \deg t_h(x) + \deg r_{h-1}(x) = \deg a(x)$$

so this has degree $< a(x)$ and this one $\geq a(x)$ so thes must be both zero and

④ $t(x)r_h(x) = t_h(x)r(x)$

Because $\deg t_h(x) + \deg r_{h-1}(x) = \deg a(x)$, $t_h \neq 0$ and by condition 1 and equation ④ $r(x) \mid r_h(x)$ but has same or higher degree so it is a scalar multiple of $r_h(x)$. Dividing the previous equation by $r_h(x)$ we see that $t(x)$ is also a scalar multiple of $t_h(x)$

Using this theorem we can find $\Lambda(x)$ and $\Omega(x)$ because they satisfy the necessary condition if we identify

$$a(x) := x^{2t}, \quad b(x) := S(x), \quad t(x) := \Lambda(x), \quad r(x) = \Omega(x)$$

The error evaluator poly. has no zeroes in common with the error locator poly. $\Lambda(x)$ bcz:

$$\Omega(\alpha^{-i_K}) = e_{i_K} \prod_{l \neq K} (1 - \alpha^{i_l - i_K}) \neq 0$$

so $\gcd(\Lambda(x), \Omega(x)) = 1$ The degree of $\Lambda(x)$ is equal to the number of errors $v \leq t$, the degree of $\Omega(x)$ is smaller as it is the sum of polynomials of degree $v-1$ and

$$\deg \Lambda(x) + \deg \Omega(x) \leq v + v - 1 \leq 2t - 1 < \deg x^{2t}$$

The constant c must be chosen such that $c t_h(0) = \Lambda(0) = 1$, we claim h is the unique index such that

$$\deg r_h < t \leq \deg r_{h-1}$$

Indeed smaller values of i would result in a poly. $\Omega(x) = c r_h(x)$ whose deg $> t-1$. On the other hand we have

$$\forall i > h \quad \deg t_i \geq \deg t_{h+1} = \deg a - \deg r_{h-1}$$

So then $\Lambda(x)$ will have a degree larger than t.

Knowing all this we can create the algorithm

Berkelamp-Massy-Fornes algorithm

1) Compute syndrom poly. $S(x)$ out of recieved word $w(x)$
2) Use Euclid's algorithm for x^{2t} and $S(x)$ to find an index h such that $\deg r_h < t \leq \deg r_{h-1}$
3) Define $\Lambda(x) = t_h(x)/t_h(0)$ and $\Omega(x) = r_h(x)/t_h(0)$.
   find the i_K such that α^{-i_K} is a root of $\Lambda(x)$
   There have to be $\deg \Lambda(x)$ distinct roots otherwise too many errors have occurred and you must ask for retransmission.
4) Define $e_{i_K} = \dfrac{-\alpha^{i_K} \Omega(\alpha^{-i_K})}{\Lambda'(\alpha^{-i_K})}$
   the corrected polynomial is $w(x) - \sum_K e_{i_K} x^{i_K}$

example: let's look at a R-S code over $\mathbb{F}_8 = \{0, 1, \alpha, \alpha^2 \dots \alpha^6\}$, where α is a zero of the minimum polynomial $\pi(x) = x^3 + x + 1$. We work modulo 2 (so $+ = -$).

For clarity this is the conversion table for additive and multiplicative notation

0	1	α	α^2	α^3	α^4	α^5	α^6
0	1	α	α^2	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	α^2+1

met $\alpha^7 = 1$.

We will create a code that corrects up to $t=2$ errors

The generator polynomial then must be of degree $2t=4$

$$g(x) = (x+1)(x+\alpha)(x+\alpha^2)(x+\alpha^3) = (x^2+\alpha^3 x+\alpha)(x^2+\alpha^5 x+\alpha^5) =$$
$$= x^4+(\alpha^3+\alpha^5)x^3+(\alpha+\alpha^5+\alpha^8)x^2+(\alpha^6+\alpha^8)x+\alpha^6 = x^4+\alpha^2x^3+\alpha^5x^2+\alpha^5x+\alpha^6$$

The codewords have length 7, the gen. poly degree 4, so the dimension is 3

Then we have a $[7, 3, 5]$ code This has $8^3 = 512$ codewords.

Any poly $a_2x^2+a_1x+a_0$ of degree=2 ($a_i \in \mathbb{F}_8$) generates a codeword (or better a code poly) after multiplication with the generator polynomial.

Now suppose we recieve the word $[\alpha, \alpha^6, \alpha^2, \alpha^4, \alpha^2, \alpha^6, \alpha]$, which corresponds to the poly:

$$r(x) = \alpha x^6+\alpha^6x^5+\alpha^2x^4+\alpha^4x^3+\alpha^2x^2+\alpha^6x+\alpha$$

Then we can calculate the $2t=4$ syndromes and determine the syndrome polynomial

- $S_0 = r(1) = \alpha+\alpha^6+\alpha^2+\alpha^4+\alpha^2+\alpha^6+\alpha = \alpha^4$.
- $S_1 = r(\alpha) = \alpha^7+\alpha^{11}+\alpha^6+\alpha^7+\alpha^4+\alpha^7+\alpha = \alpha^2+\alpha = \alpha^4$.
- $S_2 = r(\alpha^2) = \alpha^{13}+\alpha^{16}+\alpha^{10}+\alpha^6+\alpha^8+\alpha = \alpha^2$.
- $S_3 = r(\alpha^3) = \alpha^{19}+\alpha^{21}+\alpha^{14}+\alpha^{13}+\alpha^8+\alpha^8+\alpha = \alpha^2+\alpha = \alpha^4$.

Syndrome polynomial: $S(x) = \alpha^4x^3+\alpha^2x^2+\alpha^4x+\alpha^4$

we apply Euclid's algorithm to x^4 and $S(x)$

	$s_i(X)$	$t_i(X)$	$r_i(X)$
Rij_1	1	0	X^4
Rij_2	0	1	$\alpha^4X^3+\alpha^2X^2+\alpha^4X+\alpha^4 = S(X)$
$\text{Rij}_3 = \text{Rij}_1 - (\alpha^3X+\alpha)\,\text{Rij}_2$	1	$\alpha^3X+\alpha$	$\alpha X^2+\alpha^4X+\alpha^5$
$\text{Rij}_4 = \text{Rij}_2 - (\alpha^3X+\alpha^5)\,\text{Rij}_3$	$s_h(X)$	$\underbrace{\alpha^6X^2+\alpha^2X+\alpha^2}_{t_h(X)}$	$\underbrace{\alpha^6}_{r_h(X)}$

in this case $r_h(x)$ is the first poly in the right column with degree<2

the intermidiate operations were

Rij_3

$$\begin{array}{rrrr|l} X^4 & & & & \alpha^4X^3+\alpha^2X^2+\alpha^4X+\alpha^4 \\ X^4 & +\alpha^3X^3 & +X^2 & +X & \alpha^3X+\alpha \\ \hline & +\alpha^3X^3 & +X^2 & +X & \\ & +\alpha^3X^3 & +\alpha^6X^2 & +\alpha^4X & +\alpha^4 \\ \hline & & \alpha^2X^2 & +\alpha^4X & +\alpha^4 \end{array}$$

Rij_4

$$\begin{array}{rrrr|l} \alpha^4X^3 & +\alpha^2X^2 & +\alpha^4X & +\alpha^4 & \alpha X^2+\alpha^4X+\alpha^5 \\ \alpha^4X^3 & +X^2 & +\alpha X & & \alpha^3X+\alpha^5 \\ \hline & \alpha^6X^2 & +\alpha^2X & +\alpha^4 & \\ & \alpha^6X^2 & +\alpha^2X & +\alpha^3 & \\ \hline & & & \alpha^6 & \end{array}$$

$t_h(X)$ $\quad t_{i+1} = t_{i-1} - q_i t_i$

$$1-(\alpha^3X+\alpha^5)(\alpha^3X+\alpha) = 1-\alpha^6X^2+(\alpha+\alpha^4)X+\alpha^6$$
$$= \alpha^6X^2+\alpha^2X+\alpha^2 = t_h(X)$$

we can now determine $\Lambda(x)$ and $\Omega(x)$

- $\Lambda(x) = \frac{t_h(x)}{t_h(0)} = \frac{\alpha^6 x^2 + \alpha^2 x + \alpha^2}{\alpha^2} = \alpha^4 x^2 + x + 1$

we find the zeroes with brute force

$$\Lambda(1) = \alpha^4$$
$$\Lambda(\alpha) = \alpha^4$$
$$\Lambda(\alpha^2) = \alpha^5$$
$$\Lambda(\alpha^3) = 1$$
$$\Lambda(\alpha^4) = 0$$
$$\Lambda(\alpha^5) = \alpha^5$$
$$\Lambda(\alpha^6) = 0$$

$\alpha^4 = \alpha^{-3}$ and $\alpha^6 = \alpha^{-1}$

the location is given by the negative ones, so $i=1$, $i=3$

- $\Omega(x) = \frac{r_h(x)}{t_h(0)} = \frac{\alpha^6}{\alpha^2} = \alpha^4$ and $\Lambda'(x) = 2\alpha^4 x + 1 = 1$

we then get the magnitude of the error via $e_i = \frac{-\alpha^i \Omega(\alpha^{-i})}{\Lambda'(\alpha^{-i})}$

$$e_1 = \frac{\alpha\Omega(\alpha^6)}{1} = \alpha^5$$
$$e_3 = \frac{\alpha^3\Omega(\alpha^4)}{1} = \alpha^7 = 1$$

we can now reconstruct the sent polynomial

$$\begin{array}{r} \alpha x^6 + \alpha^6 x^5 + \alpha^2 x^4 + \alpha^4 x^3 + \alpha^2 x^2 + \alpha^6 x + \alpha \\ + x^3 \qquad\qquad + \alpha^5 x \\ \hline \alpha x^6 + \alpha^6 x^5 + \alpha^2 x^4 + \alpha^5 x^3 + \alpha^2 x^2 + \alpha x + \alpha \end{array}$$

we can then the original message by dividing by the generator polynomial
if we get a remainder $\neq 0$ we made a mistake

$$\begin{array}{rrrrrrr|l}
\alpha X^6 & +\alpha^6 X^5 & +\alpha^2 X^4 & +\alpha^5 X^3 & +\alpha^2 X^2 & +\alpha X & +\alpha & X^4 + \alpha^2 X^3 + \alpha^5 X^2 + \alpha^5 X + \alpha^6 \\
\alpha X^6 & +\alpha^3 X^5 & +\alpha^6 X^4 & +\alpha^6 X^3 & +X^2 & & & \alpha X^2 + \alpha^4 X + \alpha^2 \\
\hline
 & \alpha^4 X^5 & +X^4 & +\alpha X^3 & +\alpha^6 X^2 & +\alpha X & +\alpha & \\
 & \alpha^4 X^5 & +\alpha^6 X^4 & +\alpha^2 X^3 & +\alpha^2 X^2 & +\alpha^3 X & & \\
\hline
 & & \alpha^2 X^4 & +\alpha^4 X^3 & +X^2 & +X & +\alpha & \\
 & & \alpha^2 X^4 & +\alpha^4 X^3 & +X^2 & +X & +\alpha & \\
\hline
 & & & & & 0 & &
\end{array}$$

↑ matches with sent message $[\alpha^2, \alpha^4, \alpha]$