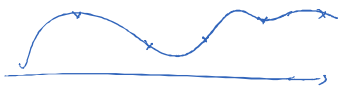
Week 8

Monday, February 29, 2016 1:26 PM

Recap

Reed - Solomon codes

→ MDS codes



$\vec{m} = (m_1, m_2, \dots m_k) \in F_q^k$

$m(x) = m_1 + m_2 x + \dots m_k x^{k-1}$

$S \to \{\alpha_1, \alpha_2 \dots \alpha_n\} \subseteq F_q - \{0\}$

$\vec{c} = (c_1, c_2 \dots c_n)$

$c_i = m(\alpha_i)$

Min distance $= n - k + 1$

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

Generalized Reed Solomon codes (GRS)

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & & \alpha_n^{k-1} \end{pmatrix} \begin{pmatrix} v_1 & 0 & \cdots & 0 \\ 0 & v_2 & \cdots & 0 \\ \vdots & & & \\ \cdots & & & v_n \end{pmatrix}$$

$v_1, v_2, \dots v_n \in F_q - \{0\}$

Generalized RS codes

conventional RS codes

n divides $q-1$.

α be an element of multiplicative order n.

$S \to \{1, \alpha, \dots \alpha^{n-1}\}$

$v_1 = v_2 = \dots v_n = 1$

primitive RS codes

$S = F_q - \{0\}$

F_7

$\{0, 1, \alpha, \dots \alpha^5\}$

$o(\alpha^2) = 3$

}

Duality

conventional Reed Solomon code

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & & \alpha^{(n-1)(k-1)} \end{pmatrix} \to$$

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{n-k-1} & \alpha^{(n-k)2} & & \alpha^{(n-1)(n-k)} \end{pmatrix} \to$$

$$(H G^T)_{(1,1)} = (1 + \alpha + \alpha^2 \dots \alpha^{n-1})$$

$$= \frac{\alpha^n - 1}{\alpha - 1}$$

$$= 0 \quad \because \alpha^n = 1$$

$n \quad i(l-1) \quad (i-1)(l-1)$

$\vec{c} = (c_1, c_2 \dots c_n)$ is a codeword

$c_1 + c_2\alpha + \dots c_n \alpha^{n-1} = 0$

$c_1 + c_2\alpha^2 \dots c_n \alpha^{2(n-1)} = 0$

More generally, $c_1 + c_2 x \dots c_n x^{n-1} = 0 \to c(x)$

if $x = \alpha, \alpha^2 \dots \alpha^{n-k}$

$g(x) = (x-\alpha)(x-\alpha^2)\dots(x-\alpha^{n-k})$

$g(x)$ divides $c(x)$

$\Rightarrow c(x) = g(x)\, u(x)$

for $\deg(u(x)) \leq k-1$

$$= 0 \quad \because \alpha^n = 1$$

$$(HG^T)_{(i,j)} = \sum_{l=1}^{n} \alpha^{i(l-1)} \cdot \alpha^{(j-1)(l-1)}$$

$$= \begin{pmatrix} 1 & \alpha^i & \alpha^{2i} & \cdots & \alpha^{(n-1)i} \end{pmatrix} \begin{pmatrix} 1 \\ \alpha^{j-1} \\ \vdots \\ \alpha^{(n-1)(j-1)} \end{pmatrix}$$

$$(HG^T)_{i,j} = \sum_{l=1}^{n} \left(\alpha^{i+j-1}\right)^{l-1} \qquad \underline{\alpha^{i+j-1} = \beta}$$

$$= \beta^0 + \beta^1 + \dots \beta^{n-1}$$

$$= \frac{\beta^n - 1}{\beta - 1} = \frac{\alpha^{n(i+j-1)} - 1}{\alpha^{i+j-1} - 1}$$

note $i + j \leq n - k + k = n$

$\Rightarrow i + j - 1 \leq n - 1$

$\Rightarrow \alpha^{i+j-1} \neq 1$

Also $\alpha^{n(i+j-1)} = 1$

$$\boxed{\Rightarrow (HG^T)_{i,j} = 0}$$

In fact For conventional Reed solomon code C.

$$C = \left\{ g(x)\, u(x) : \begin{array}{l} g(x) = (x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{n-k}) \\ \deg(u(x)) \leq k-1 \end{array} \right\}$$

g → generator polynomial

Observations

1) Duality

$$H = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{(n-1)} \\ 1 & \vdots & & \vdots \\ 1 & \alpha^{n-k} & & \alpha^{(n-1)(n-k)} \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{n-k-1} & \alpha^{2(n-k-1)} & & \alpha^{(n-1)(n-k-1)} \end{bmatrix}}_{n-k \times n} \begin{bmatrix} 1 & & & 0 \\ & \alpha & & \\ & & \alpha^2 & \\ 0 & & & \ddots \; \alpha^{n-1} \end{bmatrix} \to n \times n$$

$\Rightarrow$ Dualcode is a GRS code.
with $\alpha_i = \alpha^{i-1}$, $v_i = \alpha^{i-1}$

In fact, it can be shown that the dual code of any GRS code is also a GRS code

---

2) Fourier transform interpretation

Let $\vec{m} = \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix} \in F^n$ → field.

F. Let of $\alpha \in F$ is be the a vector root of 1

$$c_i = \sum_{j=1}^{n} m_j \alpha^{(i-1)(j-1)}$$

$$\vec{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$

Suppose I set $m_{k+1} = m_{k+2} = \dots m_n = 0$,
then I get a Reed solomon code !!

Inverse Fourier transform $(i-1)(j-1)$

then I get a Reed solomon code !!

Inverse Fourier transform.

$$m_i = \sum_{j=1}^{n} c_j \, \alpha^{-(i-1)(j-1)}$$

parity check equations are.

$$m_{k+1} = m_{k+2} = \cdots \quad m_n = 0$$

---

3) <u>Cyclic nature</u>

Take a codeword,

$$C(x) = c_1 + c_2 x + \cdots c_n x^{n-1}$$

$$= g(x) \; u(x)$$

$$\hookrightarrow (x-\alpha) \cdots (x-\alpha^{n-k})$$

consider the polynomial

$$c'(x) = c_n + c_1 x + \cdots c_{n-2} x^{n-2} + c_{n-1} x^{n-1} \rightarrow \text{cyclic shift of } \vec{c}$$

$$c'(\alpha) = c_n + c_1 \alpha + \cdots c_{n-2} \alpha^{n-2} + c_{n-1} \alpha^{n-1}.$$

$$= 0$$

$$x\,(c(x)) = c_1 x + c_2 x^2 + \cdots c_n x^n$$

$$\alpha\,(c(\alpha)) = c_1 \alpha + \cdots c_{n-1} \alpha^{n-1} + c_n \alpha^n =$$

$$0 = c_1 \alpha + \cdots c_{n-1} \alpha^{n-1} + c_n = c'(\alpha)$$

In general

$$c'(x) = 0 \text{ if } x = \alpha, \alpha^2 \cdots \alpha_{n-k}.$$

$\Rightarrow$ $\vec{c} \rightarrow$ codeword of conventional RS code $\Rightarrow C(x)$ is a codeword.

$\vec{c}' \rightarrow$ <u>cyclic shift of $\vec{c}$</u>

$\Rightarrow$ $\vec{c}' \rightarrow$ also a codeword.

cyclic codes

<u>Decoding of RS codes</u>

$\rightarrow$ Erasures

$\rightarrow$ Errors

<u>Erasures</u>

<u>Example</u> $F_8 = F_2(\alpha) / \alpha^3 + \alpha + 1$

$(7, 4, 4)$ RS code.

- $\beta \rightarrow$ primitive

$\Rightarrow$ β^n primitive if n does not divide $q-1$

in $F_8 \rightarrow \beta, \beta^2, \beta^3, \beta^4, \ldots \beta^6 \rightarrow$ all primitive

000 →	0	0
001 →	1	α^0
010 →	α	α^1
011 →	$\alpha+1$	α^3
100 →	α^2	α^2
101 →	α^2+1	α^6
110 →	$\alpha^2+\alpha$	α^4
111 →	$\alpha^2+\alpha+1$	α^5

$\alpha \rightarrow$ primitive

$m \rightarrow (0, 1, \alpha, \alpha)$ $\qquad 1, \alpha, \ldots \alpha^6$ evaluation points

$m(x) = x + \alpha x^2 + \alpha x^3$

$$\vec{c} = (1, \alpha+\alpha^3+\alpha^4, \alpha^2+\alpha^5+1, \alpha^3+1+\alpha^3, \alpha^4+\alpha^2+\alpha^6, \alpha^5+\alpha^4+\alpha^2, \alpha^6+\alpha^6+\alpha^5)$$

$$\vec{c} = (1, \alpha^5, \alpha, 1, \alpha^5, \alpha^6, \alpha^5)$$

$$\vec{c}_E = (1, \alpha^5, \alpha, 1, \underset{c_5}{*}, \underset{c_6}{*}, \alpha^5)$$

$$H \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ & \vdots & & & & & \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = 0$$

$$\begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ & & & & & & \end{bmatrix} \begin{bmatrix} 1 \\ \alpha^5 \\ \alpha \\ 1 \\ c_5 \\ c_6 \\ \alpha^5 \end{bmatrix} = 0$$

$\alpha^4 c_5 + \alpha^5 c_6 = 1+\alpha^6+\alpha^3+\alpha^3+\alpha^4 = \alpha$

$\alpha c_5 + \alpha^2 c_6 = 1+1+\alpha^5+\alpha^6+\alpha^3 = 1$

$$c_5 = \frac{\begin{vmatrix} \alpha^5 & \alpha \\ \alpha^3 & 1 \end{vmatrix}}{\begin{vmatrix} \alpha^4 & \alpha^5 \\ \alpha & \alpha^3 \end{vmatrix}} = \frac{\alpha^5-\alpha^4}{\alpha^7-\alpha^6} = \alpha^5$$

$$\vec{y} = H\vec{y} = 0 \approx O(n^3)$$

Second method

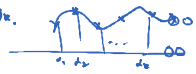
polynomial interpolation

idea, fit a degree $k-1$ polynomial that passes through any k uni-erased points

Lagrange interpolation

$\alpha_1, \alpha_2, \ldots \alpha_k \longrightarrow$ unerased co-ordinates

Receive $y_1, y_2, \ldots y_k$.



$$p_j(x) = \prod_{\substack{i=1 \\ i \neq j}}^{k} \frac{x-\alpha_i}{\alpha_j-\alpha_i}$$

$\underbrace{(x-a_1)(x-a_2)\cdots(x-a_k)}$

$$p_j(x) \longrightarrow \begin{cases} 1 & \text{if } x = \alpha_j \\ 0 & \text{if } x = \alpha_i \neq \alpha_j \end{cases}$$

$$m(x) = \sum_{i=1}^{k} y_i \, p_i(x)$$

$(k-1)$ deg poly passes through $(\alpha_1, y_1), (\alpha_2, y_2) \ldots (\alpha_k, y_k)$

Naive method $\approx O(k^2)$ steps

FFT-based method $\approx O(k \log k)$

Decoding errors

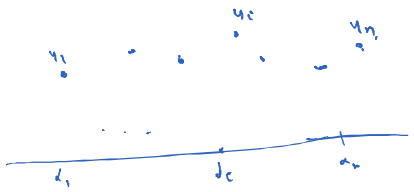
(n, k) Reed Solomon code

Transmit $\vec{x}$, Receive $\vec{y} \to 1 \times n$

$\vec{x} \to 1 \times n$

# of errors $e \leq \left\lfloor \frac{n-k}{2} \right\rfloor = \tau$.

High-level idea



Find a degree $\leq (k-1)$ polynomial, $\hat{m}(x)$ that passes through $k + \left(\frac{n-k}{2}\right)$ points

$\hat{m}(x)$ passes through at least k uncorrupted points, so $\hat{m}(x) = m(x)$

<u>Berlekamp-Welch algorithm</u>

$$P(X,Y) = E(X)\,Y - E(X)\,m(X)$$

$$E(X) = \prod_{\substack{\alpha_i \text{ is an} \\ \text{error}}} (X - \alpha_i), \quad \deg E(x) \le \tau$$

$$P(\alpha_i, y_i) = E(\alpha_i)\left[y_i - m(\alpha_i)\right]$$
$$= 0, \quad i \in \{1, 2, \dots n\}$$

<u>Goal</u>

Find poly $N_1(x)$, $E_1(x)$, using $(\alpha_1, y_1) \dots (\alpha_n, y_n)$

1) $Q(X,Y) = E_1(X)\,Y - N_1(X)$

   $Q(\alpha_i, y_i) = 0 \quad \forall i = 1, 2, \dots n.$

2) $\deg(E_1(x)) \le \tau$

3) $\deg(N_1(x)) \le k + \tau - 1$ $\longrightarrow$ Suppose, # of errors $\le \tau$.

<u>Properties</u> $\longrightarrow$ $Q(X,Y)$ satisfying 1), 2), 3)

exists

Yes, set $E_1(x) = E(x)$

$N_1(x) = m(x)\,E(x)$

$\longrightarrow$ Any $Q(X,Y)$ satisfying 1), 2), 3)

satisfies, $\dfrac{N_1(x)}{E_1(x)} = m(x)$