Lecture 7: MacWilliams identities. Reed–Solomon codes.

Course instructor: Alexey Frolov

al.frolov@skoltech.ru

Teaching Assistant: Stanislav Kruglik

stanislav.kruglik@skolkovotech.ru

February 14, 2017

Outline

Outline

Weight spectrum and enumerator

Definition

Let $\mathcal{C}$ be a linear (n, k, d) code. A vector $A(\mathcal{C}) = [A_0, A_1, \ldots, A_n]$, where

$$A_W = |\{c \in \mathcal{C} : ||c|| = W\}|.$$

is called a weight spectrum of $\mathcal{C}$.

Definition

Let $A(\mathcal{C}) = [A_0, A_1, \ldots, A_n]$ be a weight spectrum of a code $\mathcal{C}$, then

$$W_{\mathcal{C}}(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i = \sum_{c \in \mathcal{C}} x^{n-||c||} y^{||c||}$$

is called a weight enumerator of $\mathcal{C}$.

Example

Example

$\mathcal{C} = \{000, 011, 101, 110\}$

$$W_{\mathcal{C}}(x, y) = x^3 + 3xy^2$$

.

MacWilliams identities

Theorem (F. J. MacWilliams, 1963))

$$W_{\mathcal{C}^\perp}(x,y) = \frac{1}{|\mathcal{C}|} W_{\mathcal{C}}(x+y, x-y).$$

Hadamard transform

Definition (Dot product)

Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$:

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 \oplus \ldots x_n y_n.$$

Definition (Hadamard transform)

Let f be an arbitrarily function defined on $\mathbb{F}_2^n$. Then

$$\hat{f}(\mathbf{x}) = \sum_{\mathbf{a} \in \mathbb{F}_2^n} (-1)^{\mathbf{a} \cdot \mathbf{x}} f(\mathbf{a}).$$

is called a Hadamard transform of f.

Lemma

$$\sum_{\mathbf{x} \in \mathcal{C}^\perp} f(\mathbf{x}) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{x} \in \mathcal{C}} \hat{f}(\mathbf{x})$$

Proof of lemma

$$\sum_{\mathbf{x}\in\mathcal{C}} \hat{f}(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{C}} \sum_{\mathbf{a}\in\mathbb{F}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}} f(\mathbf{a}) =$$

$$= \sum_{\mathbf{a}\in\mathbb{F}_2^n} \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} f(\mathbf{a}) = \sum_{\mathbf{a}\in\mathbb{F}_2^n} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} =$$

$$= \sum_{\mathbf{a}\in\mathcal{C}^\perp} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} + \sum_{\mathbf{a}\notin\mathcal{C}^\perp} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}}$$

1. $\forall\, \mathbf{a} \in \mathcal{C}^\perp, \mathbf{x} \in \mathcal{C}: \quad \mathbf{a}\cdot\mathbf{x} = 0 \;\Rightarrow$

$$\sum_{\mathbf{a}\in\mathcal{C}^\perp} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} = \sum_{\mathbf{a}\in\mathcal{C}^\perp} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} 1 = |\mathcal{C}| \sum_{\mathbf{a}\in\mathcal{C}^\perp} f(\mathbf{a})$$

2. $\forall\, \mathbf{a} \notin \mathcal{C}^\perp$: «$\mathbf{a}\cdot\mathbf{x} = 0$» и «$\mathbf{a}\cdot\mathbf{x} = 1$» occur equal times in the second sum $\Rightarrow \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} = 0 \;\Rightarrow$

$$\sum_{\mathbf{a}\notin\mathcal{C}^\perp} f(\mathbf{a}) \sum_{\mathbf{x}\in\mathcal{C}} (-1)^{\mathbf{a}\cdot\mathbf{x}} = 0$$

□

Proof of theorem

Consider

$$f(\mathbf{b}) = x^{n-\|\mathbf{b}\|} y^{\|\mathbf{b}\|}$$

Apply Hadamard transform

$$\hat{f}(\mathbf{a}) = \sum_{\mathbf{b}\in\mathbb{F}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{b}} f(\mathbf{b}) = \sum_{\mathbf{b}\in\mathbb{F}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{b}} x^{n-\|\mathbf{b}\|} y^{\|\mathbf{b}\|} =$$

(Note, that $y^{\|\mathbf{b}\|} = y^{b_1} \dots y^{b_n}$ и $x^{n-\|\mathbf{b}\|} = x^{1-b_1} \dots x^{1-b_n}$)

$$= \sum_{\mathbf{b}\in\mathbb{F}_2^n} (-1)^{a_1 b_1 + \dots + a_n b_n} \prod_{i=1}^{n} x^{1-b_i} y^{b_i} =$$

$$= \sum_{b_1=0}^{1} \sum_{b_2=0}^{1} \dots \sum_{b_n=0}^{1} \prod_{i=1}^{n} (-1)^{a_i b_i} x^{1-b_i} y^{b_i} =$$

$$= \prod_{i=1}^{n} \sum_{w=0}^{1} (-1)^{a_i w} x^{1-w} y^{w} =$$

Proof of theorem

$$\hat{f}(\mathbf{a}) = \ldots = \prod_{i=1}^{n} \sum_{w=0}^{1} (-1)^{a_i w} x^{1-w} y^{w} =$$

$$= (x+y)^{n-\|\mathbf{a}\|}(x-y)^{\|\mathbf{a}\|},$$

as

$$\sum_{w=0}^{1} (-1)^{a_i w} x^{1-w} y^{w} = \begin{cases} x+y, & \text{если } a_i = 0, \\ x-y, & \text{если } a_i = 1. \end{cases}$$

According to lemma $$\frac{1}{|\mathcal{C}|} \sum_{\mathbf{a} \in \mathcal{C}} \hat{f}(\mathbf{a}) = \sum_{\mathbf{a} \in \mathcal{C}^{\perp}} f(\mathbf{a}),$$

thus $$\frac{1}{|\mathcal{C}|} \sum_{\mathbf{a} \in \mathcal{C}} (x+y)^{n-\|\mathbf{a}\|}(x-y)^{\|\mathbf{a}\|} = \sum_{\mathbf{a} \in \mathcal{C}^{\perp}} x^{n-\|\mathbf{a}\|} y^{\|\mathbf{a}\|}.$$ □

Example

Example

Let $\mathbf{H} = [11\ldots1]$ be a parity check matrix of single parity check (SPC) code with length n. Find its enumerator.
Note, that the dual code is a repetition code with

$$W_{\mathcal{C}^\perp}(x, y) = x^n + y^n,$$

thus we have

$$W_{\mathcal{C}}(x, y) = \frac{1}{|\mathcal{C}^\perp|}\left[(x+y)^n + (x-y)^n\right] = \frac{1}{2}\left[(x+y)^n + (x-y)^n\right].$$

Outline

MDS code

Definition

An (n, k, d) code is called Maximum Distance Separable (MDS) code if

$$d = n - k + 1.$$

This means, that the code meets the Singleton bound.

Properties of MDS codes

Proposition

A q-ary $[n, k]$ linear code is an MDS code precisely if the parity check matrix $\mathbf{H}$ has every set of $n - k$ columns linearly independent.

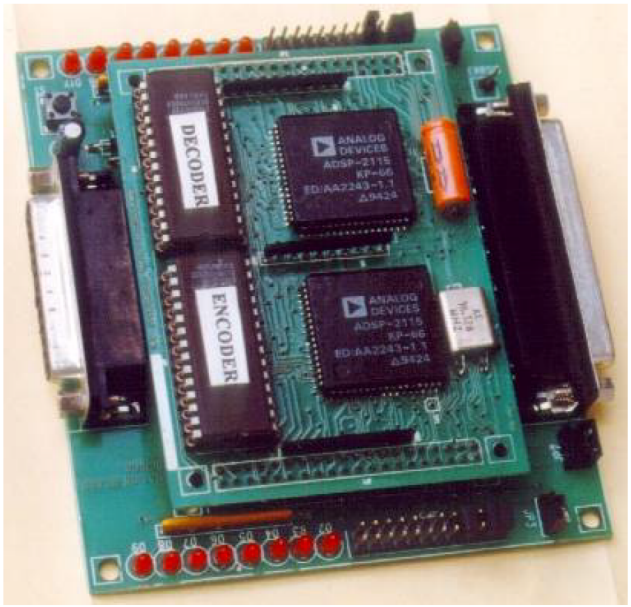
Proposition

The code $\mathcal{C}^{\perp}$ dual to $\mathcal{C}$ is a linear MDS code if $\mathcal{C}$ itself is a linear MDS code.

Outline

The best algebraic codes

Millions of error-correcting codes are decoded **every minute**, with efficient algorithms implemented in custom VLSI circuits.



At least 50% of these VLSI circuits **decode Reed-Solomon codes.**

I.S. Reed and G. Solomon, Polynomial codes over certain finite fields, *Journal Society Indust. Appl. Math.* **8**, pp. 300-304, June 1960.

RS codes are used in ...

- ***Magnetic recording*** (all computer hard-disks use RS codes)
- ***Digital video broadcasting*** (ETSI DVB-T and DVB-H)
- ***Digital versatile disks*** (DVDs use products of RS codes)
- ***Third generation (3G) wireless telephony*** (IS-2000, Release D)
- ***Optical fiber networks*** (ITU–T G.795)
- ***Compact disks*** (use cross-interleaved shortened RS codes)
- ***ADSL transceivers*** (ITU–T G.992.1)
- ***Wireless broadband systems – wireless MAN*** (IEEE 802.16)
- ***Intelsat Earth stations*** (IESS–308)
- ***Space telemetry systems*** (CCSDS)
- ***Digital satellite broadcast*** (ETS 300-421S, ETS 300-429)
- ***Frequency-hop communications*** (primarily military)
- ***Deep-space exploration*** (all NASA probes)

RS codes: generator matrix view

Let $\mathbb{F}_q$ be the finite field of order q and let $\mathbb{F}_q[x]$ denote the ring of polynomials over $\mathbb{F}_q$ in the variable x. Given a set $\mathcal{B}$ of n pairwise different field elements

$$\mathcal{B} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$$

from $\mathbb{F}_q$.

The Reed-Solomon code $RS(n, k)$, $1 \leq k \leq n$, is defined as follows

$$RS(n, k) = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) : f(x) \in \mathbb{F}_q[x], \deg f(x) < k\}.$$

The Reed-Solomon code is a linear code over $\mathbb{F}_q$. It has length $n \leq q$, dimension k and $d = n - k + 1$.

RS codes: distance and generator matrix

The polynomial of degree $\leq k-1$ cannot have more than $k-1$ roots (zeroes), thus

$$d = n - k + 1.$$

Generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ {\alpha_1}^{k-1} & {\alpha_2}^{k-1} & \dots & {\alpha_n}^{k-1} \end{bmatrix}$$

Encoding

$$c = f\mathbf{G},$$

where $f = (f_0, f_1, \dots, f_{k-1})$ – vector of coefficients of $f(x)$.

RS codes: parity check matrix view

Generator polynomial:

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{d-1}).$$

Encoding:

$$c(x) = f(x)g(x).$$

Parity check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{d-1} & \dots & \alpha^{(d-1)(n-1)} \end{bmatrix}$$

What is the connection in between two points of view?

Discrete Fourier Transform

$$F = [\alpha^{ij}]_{i=0,\ldots,n-1}^{j=0,\ldots,n-1} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(n-1)} \\ 1 & \vdots & \vdots & \ldots & \vdots \\ 1 & \alpha^{n-1} & \alpha^{2(n-1)} & \ldots & \alpha^{(n-1)(n-1)} \end{bmatrix}$$

Discrete Fourier Transform

Information:

$$f = [f_0 f_1 \dots f_{k-1} 0 \dots 0]$$

Apply DFT to obtain a codeword

$$c = Ff$$

$$\begin{aligned} c(\alpha^{-k}) &= c(\alpha^{n-k}) = c(\alpha^{d-1}) = 0 \\ c(\alpha^{-k-1}) &= c(\alpha^{n-k-1}) = c(\alpha^{d-2}) = 0 \\ \vdots &= \vdots \\ c(\alpha^{-(n-1)}) &= c(\alpha) = 0 \end{aligned}$$

Outline

Notations

Let us consider a situation when t errors $\{e_{j_1}, e_{j_2}, \ldots, e_{j_t}\}$.
We introduce a notation of error locator

$$X_i = \alpha^{e_{j_i}}, \quad i = 1, \ldots, t.$$

and error values $Y_i = e_{j_i}$, $i = 1, \ldots, t$.
Let $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$. The syndrome can be calculated as follows

$$\begin{aligned} S_1 &= Y_1 X_1 + Y_2 X_2 + \ldots + Y_t X_t \\ S_1 &= Y_1 X_1^2 + Y_2 X_2^2 + \ldots + Y_t X_t^2 \\ &\ldots \\ S_1 &= Y_1 X_1^t + Y_2 X_2^t + \ldots + Y_t X_t^t \end{aligned}$$

Polynomials

Syndrome polynomial

$$S(z) = \sum_{j=1}^{2t} S_j z^{j-1}$$

Error locator polynomial

$$\sigma(z) = \prod_{i=1}^{t} (X_i z - 1)$$

Error value polynomial

$$\omega(z) = \sum_{i=1}^{t} Y_i X_i \prod_{l=1, l \neq i}^{t} (X_l z - 1).$$

Additional (unnamed) polynomial

$$\Phi(z) = \sum_{i=1}^{t} Y_i X_i^{2t+1} \prod_{l=1, l \neq i}^{t} (X_l z - 1).$$

Key equation

$$S(z)\sigma(z) = z^{2t}\Phi(z) - \omega(z)$$

To solve the equation use extended Euclidean algorithm. Start with polynomial z^{2t} and $S(z)$, stop when the degree of residue is less or equal $t-1$ for the first time. Use extended Euclidean algorithm to find $\sigma(z)$ and $\omega(z)$

Chien search

We know $\sigma(z)$, find X_i by exhaustive search over all the elements of $\mathbb{F}_q$.

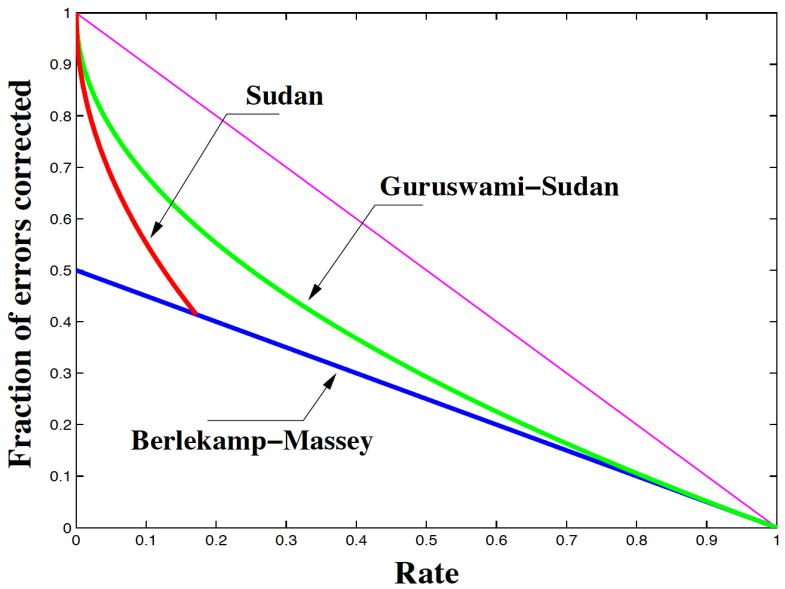
Forney's algorithm

$$Y_i = \frac{\omega(X_i^{-1})}{\sigma'_z(X_i^{-1})} \quad i = 1, \ldots, t.$$

Outline

Can we do better?

The 2002 Nevanlinna Prize went to M. Sudan with the citation "*...in the theory of error-correcting codes, he showed that certain coding methods could correct* ***many more errors*** *than previously* ***thought possible***."



M. Sudan, Decoding of Reed-Solomon codes beyond the error correction bound, *Journal of Complexity*, 1997.

V. Guruswami and M. Sudan, Improved decoding of Reed-Solomon and algebraic-geometric codes, *IEEE Trans. Information Theory*, 1999.

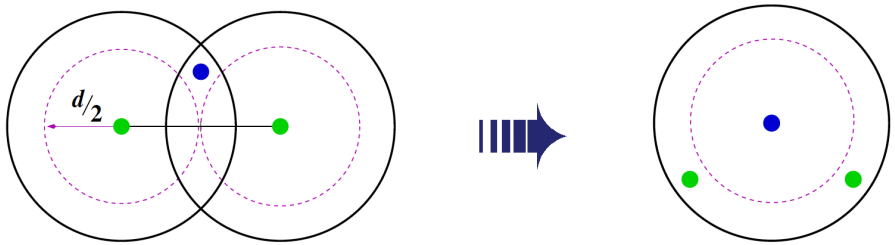
Johnson bound

List size is polynomial up to

$$\tau_J = n - \sqrt{n(n-d)}.$$

List decoding

If we attempt to correct more than $d/2$ errors by increasing the decoding radius, the decoding sphere could contain more than one codeword:



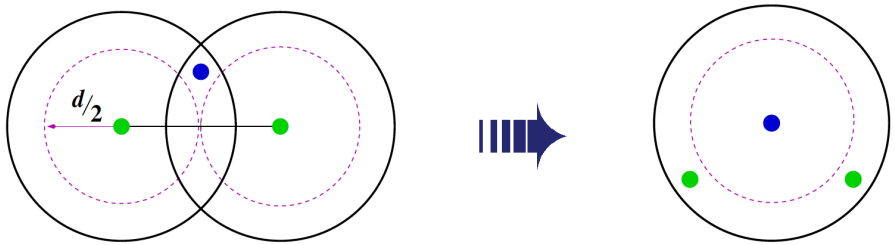
For this reason, conventional wisdom asserts that it is not possible to correct more than $d/2$ errors with a code of distance d.

List decoding: if there is more than one codeword in the decoding sphere, output the list of all of them!

- **In theory:** as long as one can guarantee that the size L of the list is small (say, constant), this is good enough for most purposes.
- **In practice:** the probability of having a list of size $\geqslant 2$ is extremely low (less that 10^{-30} for long high-rate Reed-Solomon codes).

List decoding

If we attempt to correct more than $d/2$ errors by increasing the decoding radius, the decoding sphere could contain more than one codeword:



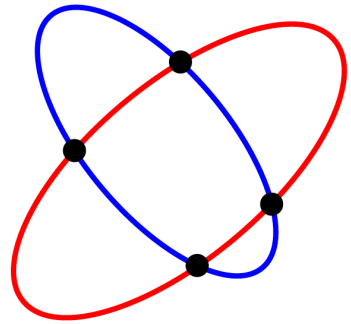
For this reason, conventional wisdom asserts that it is not possible to correct more than $d/2$ errors with a code of distance d.

List decoding: if there is more than one codeword in the decoding sphere, output the list of all of them!

- **In theory:** as long as one can guarantee that the size L of the list is small (say, constant), this is good enough for most purposes.
- **In practice:** the probability of having a list of size $\geqslant 2$ is extremely low (less that 10^{-30} for long high-rate Reed-Solomon codes).

List decoding

Every codeword of the Reed-Solomon code $\mathbb{C}_q(n,k)$ corresponds to a polynomial. The **unknown** transmitted codeword can be represented by the algebraic curve $Y - f(X)$ of degree at most $k-1$.



Bézout's Theorem

*Two algebraic curves of degrees d and δ intersect in δd points, and cannot meet in more than δd points unless the equations defining them **have a common factor.***

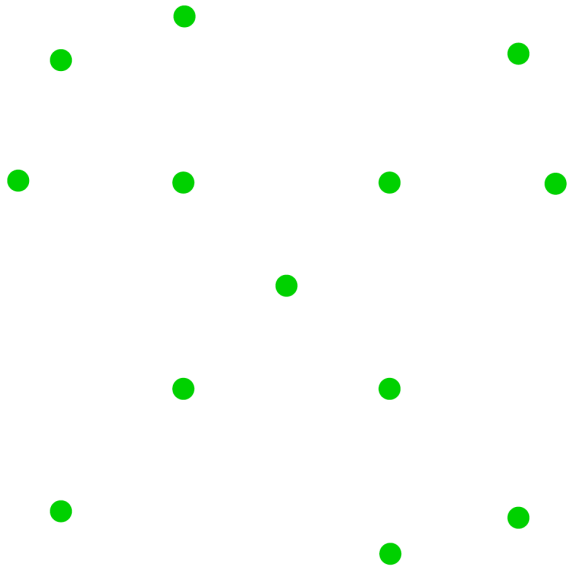
E. Bézout, *Théorie générale des équations algébriques*, Paris, 1779.

Application of Bézout's theorem for decoding

If we could construct $\mathcal{Q}(X,Y) \in \mathbb{F}_q[X,Y]$ which defines a curve of degree δ that intersects $Y - f(X)$ in more than $(k-1)\delta$ points (including points at ∞), then $Y - f(X)$ **can be recovered as a factor of** $\mathcal{Q}(X,Y)$!

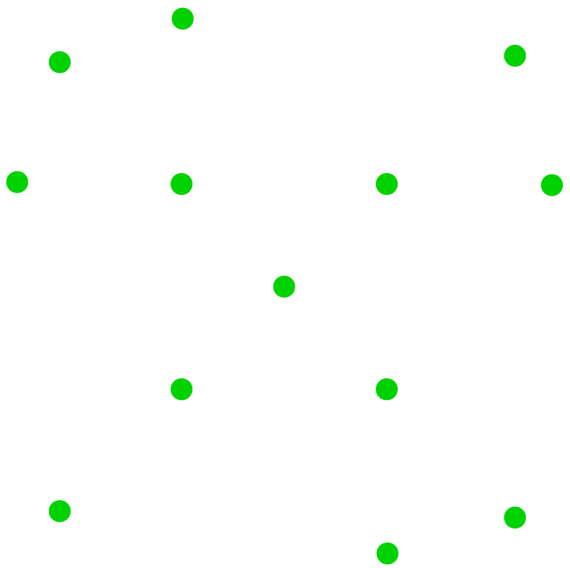
List decoding

Suppose $k = 2$, so that the Reed-Solomon codewords are lines $f(X) = aX + b$. Given 14 interpolation points, we want to compute all lines passing through at least 5 of these points.



List decoding

Suppose $k = 2$, so that the Reed-Solomon codewords are lines $f(X) = aX + b$. Given 14 interpolation points, we want to compute all lines passing through at least 5 of these points.

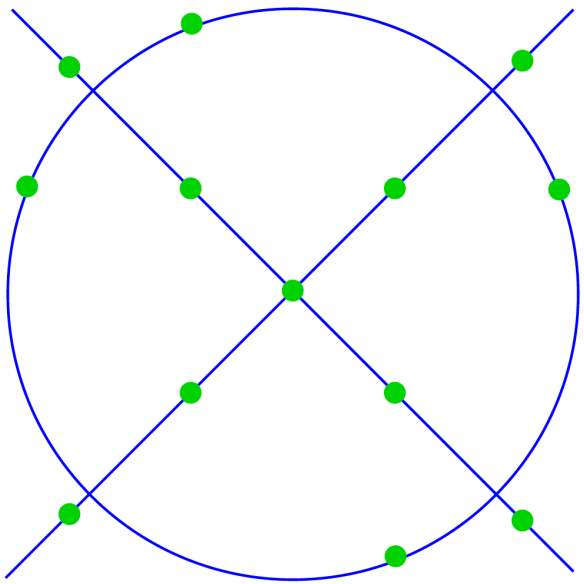


Compute a polynomial $\mathcal{Q}(X, Y)$ of degree < 5 such that $\mathcal{Q}(\alpha_i, \beta_i) = 0$ for all the 14 points:

$$\mathcal{Q}(X, Y) = Y^4 - X^4 - Y^2 + X^2$$

List decoding

Suppose $k = 2$, so that the Reed-Solomon codewords are lines $f(X) = aX + b$. Given 14 interpolation points, we want to compute all lines passing through at least 5 of these points.



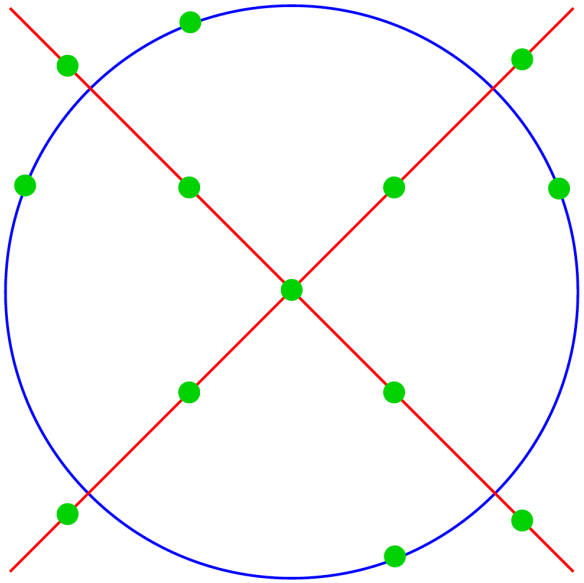
Compute a polynomial $Q(X, Y)$ of degree < 5 such that $Q(\alpha_i, \beta_i) = 0$ for all the 14 points:

$$Q(X, Y) = Y^4 - X^4 - Y^2 + X^2$$

Let's plot all the zeros of $Q(X, Y)$. All the relevant lines now emerge!

List decoding

Suppose $k = 2$, so that the Reed-Solomon codewords are lines $f(X) = aX + b$. Given 14 interpolation points, we want to compute all lines passing through at least 5 of these points.



Compute a polynomial $Q(X, Y)$ of degree < 5 such that $Q(\alpha_i, \beta_i) = 0$ for all the 14 points:

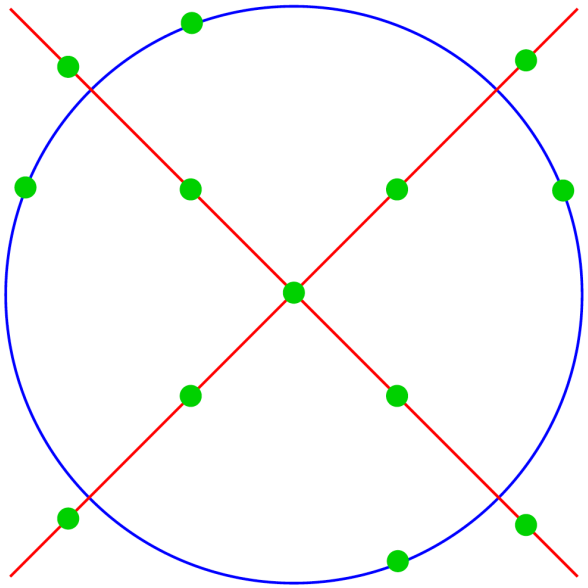
$$Q(X, Y) = Y^4 - X^4 - Y^2 + X^2$$

Let's plot all the zeros of $Q(X, Y)$. All the relevant lines now emerge! Formally, $Q(X, Y)$ factors as:

$$(Y + X)(Y - X)(X^2 + Y^2 - 1)$$

List decoding

Suppose $k = 2$, so that the Reed-Solomon codewords are lines $f(X) = aX + b$. Given 14 interpolation points, we want to compute all lines passing through at least 5 of these points.



Compute a polynomial $Q(X, Y)$ of degree < 5 such that $Q(\alpha_i, \beta_i) = 0$ for all the 14 points:

$$Q(X, Y) = Y^4 - X^4 - Y^2 + X^2$$

Let's plot all the zeros of $Q(X, Y)$. All the relevant lines now emerge! Formally, $Q(X, Y)$ factors as:

$$(Y + X)(Y - X)(X^2 + Y^2 - 1)$$

Bézout's Theorem says it must be so, since $\deg Q \times \deg f = 4$ is strictly less than the number of intersection points, which is 5.

Polynomial Reconstruction

Given parameters n, t, k and n pairs $(\alpha_i, y_i) \in \mathbb{F}_q^2$, we want to find all degree $\leq k$ polynomials $f \in F[X]$ such that

$$|\{i : f(\alpha_i) = y_i\}| \geq t$$

Sudan's algorithm

Armed with the above intuition, consider the following algorithm:

- Find a low degree $Q(X, Y) \neq 0 \in \mathbb{F}_q[X, Y]$, such that $Q(\alpha_i, y_i) = 0$ for all i.
- Find all factors of the form $yf(x)$ of $Q(x, y)$. Output those with agreement at least t.

$(1, k)$-weighted degree

Definition

For a polynomial $Q(X, Y) \in \mathbb{F}_q[X, Y]$, its $(1, k)$-weighted degree is defined to be the maximum value of $i + kj$ over all non-zero monomials $x^i y^j$ of Q.

How to find $Q(X, Y)$

Lemma

Given positive integer D and $kn < \binom{D+2}{2}$ points $\{(\alpha_i, y_i)\}_{i=1}^{n}$, the following holds: There exists $Q(X, Y) \neq 0$, such that $Q(\alpha_i, y_i) = 0$ for all i and $(1, k)$-weighted degree of Q is D. Moreover such Q can be found by solving a linear system.

Proof.

$$\begin{aligned}\sum_{j=0}^{\lfloor D/k \rfloor} (D - jk + 1) &= (D+1)\left(\left\lfloor \frac{D}{k} \right\rfloor + 1\right) - \frac{k}{2}\left\lfloor \frac{D}{k} \right\rfloor \left(\left\lfloor \frac{D}{k} \right\rfloor + 1\right) \\ &\geq \binom{D+2}{2}.\end{aligned}$$

Find $f(x)$

Lemma

Given a degree-k polynomial $f(x)$ with $f(\alpha_i) = y_i$ for t different points $(\alpha_i, y_i) : Q(\alpha_i, y_i) = 0$, if $(1, k)$-weighted degree of Q is $< t$, then $yf(x)|Q(x, y)$.

Proof.

$R(x) = Q(x, f(x))$. $R(x)$ has t different roots, at the same time the degree of $R(x) < t$.

Theorem

Given parameters n, t, k and n pairs (α_i, y_i), if $t > \sqrt{2kn}$, we can find all degree $\leq k$ polynomials $f \in \mathbb{F}_q[X]$ such that

$$|\{i : f(\alpha_i) = y_i\}| \geq t,$$

in time polynomial in n, k and $|\mathbb{F}_q|$. Moreover there are at most $\sqrt{2n/k}$ such fs.

Proof.

$D \leftarrow t - 1$, then $\binom{D+1}{2} = \frac{t^2+2}{2k} > n$. □

Thank you for your attention!