UNIVERSITÀ
DI TRENTO

Coding Theory and Commutative Algebra

Giacomo Borin

Università di Trento

April 8, 2024

Table of contents I

Table of contents II

Coding Theory

Coding Theory in Plain English

When we send messages on a disturbed channel it is possible that one or more errors occours, thus we would like to be able to correct them.

For example if I sent you the message:
ATTAXK THE ENEMUES AT DAWB
you will be able to recover the original message.

This happens because the english words bring a quantity of redundant information (in fact not every characters combination is an english word).

Idea

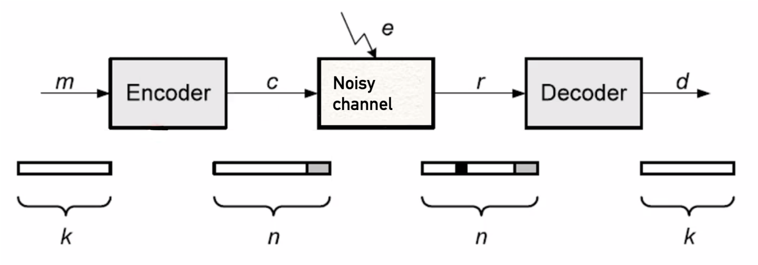


Figure: Example idea of Error correcting codes

Definition (Linear code)

A linear $[n, k]$-code is an injective linear map:

$$\mathcal{C}(n,k) : \mathbb{F}_q^k \to \mathbb{F}_q^n$$

This map is uniquely identified by the linear subspace of the image in $\mathbb{F}_q^n$, thus we call **codewords** the vectors of the image.
Sometimes to define the linear code we consider only a subspace $\mathcal{C}$ of dimension k in $\mathbb{F}_q^n$.

Using this map we can add $n - k$ bits of redundant information to the input string. The matrix G that represents the linear code is called **Generator matrix**.

Dual code

We can also associate an $n - k \times n$ matrix H called **Parity-Check matrix**, that contains the equations of the linear code.

The parity check matrix can also be seen as the generator matrix of the dual code, i.e.

Definition

Given an $[n, k]$ code $\mathcal{C}$ we can define the **dual code** $\mathcal{C}^{\perp}$ as the ortogonal space to $\mathcal{C}$

Example I

For example if we want to send a 2 bit message and correct at least one error we can use this linear code:

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

thus we `encode` the 2 bit strings as:

$$\begin{aligned} (0,0) &\mapsto (0,0,0,0,0) \\ (0,1) &\mapsto (1,0,1,1,1) \\ (1,0) &\mapsto (0,1,1,0,1) \\ (1,1) &\mapsto (1,1,0,1,0) \end{aligned}$$

Distance

Definition (Hamming Distance)

The distance of two points is the number of different coordinates:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \#\{i \mid \boldsymbol{x}_i \neq \boldsymbol{y}_i\}$$

For example

$$d((0,0,1,0,1),(0,1,1,0,0)) = 2$$

We define the minimum distance of a linear code the minimum Hamming distance between any two codewords.

Example II

To have an idea of what's happening we use graphs.

Here vertices will represents strings and the vertices will be connected if the strings have Hamming distance 1 (we can pass from one to another with one flip).

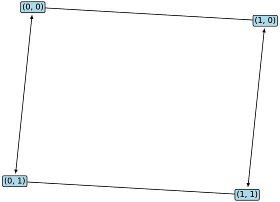


Figure: Representation of $\mathbb{F}_2^2$

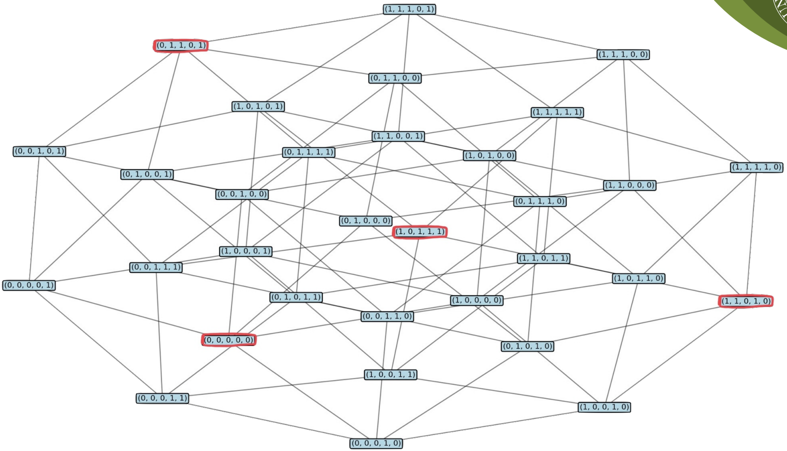


Figure: Immersion of $\mathbb{F}_2^2$ in $\mathbb{F}_2^5$

Algorithm description

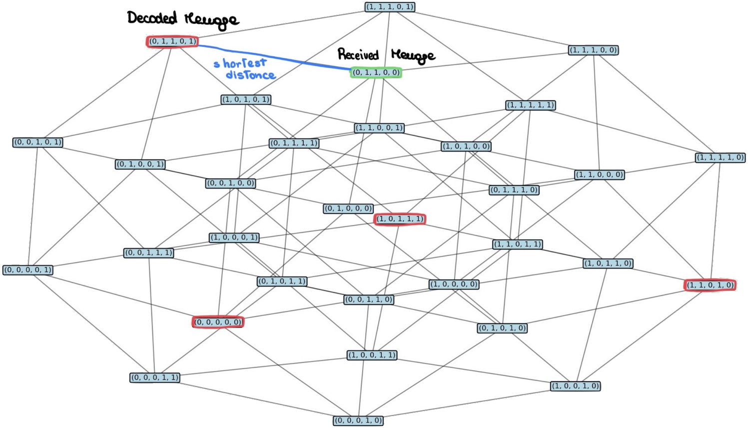
1. The first phase consist in the encoding: we add information to a k bit string through a matrix, obtaining a codeword $\boldsymbol{c}$.
2. Then the message is sent over a noisy channel, if $\boldsymbol{r}$ is the recived codeword we assume that

   $$\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$$

   where $\boldsymbol{e}$ is the error occorred.
3. The decoding algorithm is then able to invert a fixed number of errors looking for the nearest codeword.

We can see that if d is the minimum distance, then we can correct t errors if $t \leq 2d - 1$.

Suppose that we want to send $(0, 1)$. We encode it as $(0, 1, 1, 0, 1)$, but then $(0, 1, 1, 0, 0)$ is received.



Decoded Message

shortest distance

Received Message

Cyclic codes

Definition(s)

There are several way to define cyclic codes, some better than others. A simple one is

Definition

A code $\mathcal{C}$ over $\mathbb{F}_q$ is said **cyclic** if it is closed with respect to the shift operator

It is possible to have another one more interesting and algebraic.

Algebraic definition

Consider the ring:

$$\mathbb{C}_{q,n} := \frac{\mathbb{F}_q[x]}{x^n - 1}$$

We can associate an element $\boldsymbol{c} = (c_0, ..., c_{n-1}) \in \mathbb{F}_q^n$ to a polynomial

$$c_0 + c_1 x + ... + c_{n-1}\ x^{n-1} \in \mathbb{C}_{q,n}$$

And so we can also define:

Definition

A code is said **cyclic** if it can be associated (using previous association) to an ideal $I \subseteq \mathbb{C}_{q,n}$

Algebraic definition

Consider the ring:

$$\mathbb{C}_{q,n} := \frac{\mathbb{F}_q[x]}{x^n - 1}$$

We can associate an element $\boldsymbol{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ to a polynomial

$$c_0 + c_1 x + \ldots + c_{n-1}\ x^{n-1} \in \mathbb{C}_{q,n}$$

And so we can also define:

Definition

A code is said **cyclic** if it can be associated (using previous association) to an ideal $I \subseteq \mathbb{C}_{q,n}$

There is another one very simple that emphasizes the algebraic structure used.

Consider the splitting field $\mathbb{F} := \mathbb{F}_{q^m}$ of $x^n - 1 \in \mathbb{F}_q[x]$ and $\xi \in \mathbb{F}$ a primitive n-th root.

Define a subset $C = \{i_1, ..., i_r\} \subset \{1, ..., n\}$, called defining set.

Definition

The cyclic code associated to C is:

$$\mathcal{C} = \{\ c(x) \in \mathbb{C}_{q,n} |\, c(\xi^i) = 0 \text{ for all } i \in C\}$$

A defining set is said *complete defining set* of $\mathcal{C}$ if it is the maximal that defines the code.

There is another one very simple that emphasizes the algebraic structure used.

Consider the splitting field $\mathbb{F} := \mathbb{F}_{q^m}$ of $x^n - 1 \in \mathbb{F}_q[x]$ and $\xi \in \mathbb{F}$ a primitive n-th root.

Define a subset $C = \{i_1, ..., i_r\} \subset \{1, ..., n\}$, called defining set.

Definition

The cyclic code associated to C is:

$$\mathcal{C} = \{\ c(x) \in \mathbb{C}_{q,n} |\, c(\xi^i) = 0 \text{ for all } i \in C\}$$

A defining set is said *complete defining set* of $\mathcal{C}$ if it is the maximal that defines the code.

Definition

Remark

Let $C = \{i_1, ..., i_r\} \subset \{1, ..., n\}$ be a complete defining set of a code $\mathcal{C}$. Then a possible form for the Parity-Check matrix is:

$$H = \begin{bmatrix} 1 & \xi^{i_1} & \xi^{2i_1} & \dots & \xi^{(n-1)i_1} \\ 1 & \xi^{i_2} & \xi^{2i_2} & \dots & \xi^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{i_r} & \xi^{2i_r} & \dots & \xi^{(n-1)i_r} \end{bmatrix} \tag{1}$$

Decoding and syndromes

Syndromes

Given a received word $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ we can evaluate the **syndrome** of it by applying the matrix H:

$$\boldsymbol{s}^T := H\boldsymbol{r}^T = H(\boldsymbol{c} + \boldsymbol{e})^T = H\boldsymbol{c}^T + H\boldsymbol{e}^T = H\boldsymbol{e}^T$$

This can be seen also in polynomial form as :

$$s_i = s(\xi^i) := (r)(\xi^i) = (c + e)(\xi^i) = c(\xi^i) + e(\xi^i) \overset{*}{=} e(\xi^i)$$

where $*$ holds for indexes in the defining set C of the code.

So to recap syndromes can be evaluated using the received polynomial, but depends only on the error vector. Suppose now that less then equal t errors occurred, so we have:

$$\boldsymbol{e} = (0, ..., 0, e_{j_1}, 0, ..., 0, e_{j_l}, 0, ..., 0, e_{j_t})$$

thus j_l are the error positions and e_{j_l} the values.

At polynomial level for $i \in C$ we have:

$$s_i = r(\xi^i) = e(\xi^i) = \sum_{l=1}^{t} e_{j_l} \cdot (\xi^i)^{j_l} = \sum_{l=1}^{t} e_{j_l} \cdot (\xi^{j_l})^i \qquad (2)$$

So to recap syndromes can be evaluated using the received polynomial, but depends only on the error vector. Suppose now that less then equal t errors occurred, so we have:

$$\boldsymbol{e} \ = (0, ..., 0, e_{j_1}, 0, ..., 0, e_{j_l}, 0, ..., 0, e_{j_t})$$

thus j_l are the error positions and e_{j_l} the values.

At polynomial level for $i \in C$ we have:

$$s_i = r(\xi^i) = e(\xi^i) = \sum_{l=1}^{t} \ e_{j_l} \cdot (\xi^i)^{j_l} = \sum_{l=1}^{t} e_{j_l} \cdot (\xi^{j_l})^i \qquad (2)$$

Working envirorment

For notation simplicity index C as $\{i_1, ..., i_r\}$ where $r = n - k$. Consider the polynomial ring

$$\mathbb{F}_q[x_1, ..., x_r, z_1, ..., z_t, y_1, ..., y_t]$$

Here we have that:

- x_u represents the syndromes
- z_l represents the error positions, in fact $z_l = \xi^{j_l}$
- y_l represents the error values

Naive Syndromes variety

Remark

With the previous notation the equation 2 can be written as:

$$0 = \sum_{l=1}^{t} y_l \cdot (z_l)^{i_u} - x_u =: f_u \tag{3}$$

for $u \in \{1, \ldots, r\}$.

So if we substitute x_u with the known syndromes we have that the error positions and values are points of the variety

$$\mathcal{V}(f_u(s_{i_1}, \ldots, s_{i_r}), u \in \{1, \ldots, r\}) \subset \mathbb{F}_q^{2r} \tag{4}$$

We need to add relation to our variety:

1. The syndromes lie in $\mathbb{F}_{q^m}$, so we add

$$\chi_u := x_u^{q^m} - x_u$$

2. The error locations are zeros or n-th root of unity , so we add

$$h_l := z_l^{n+1} - z_l$$

3. The error values are in $\mathbb{F}_q \setminus \{0\}$, so we add

$$\lambda_l := y_l^q - 1$$

CHRT-syndrome variety

Consider the collection of polynomials:

$$F_{\mathcal{C}} = \{\ f_u, \chi_u, h_l, \lambda_l \text{ for } \ 1 \leq u \leq r, 1 \leq l \leq t\} \tag{5}$$

Definition

The zero-dimensional ideal $I_{\mathcal{C}}$ generated by $F_{\mathcal{C}}$ is called CHRT-syndrome ideal associated to the code $\mathcal{C}$, and the variety $\mathcal{V}(F_{\mathcal{C}})$ defined by $F_{\mathcal{C}}$ is called a CHRT-syndrome variety, after Chen, Reed, Helleseth and Truong ([Che+94b; Che+94c; Che+94a]).

Groebner Basis

Monomial order

Definition

Consider a *total order* $\prec$ on $\mathbb{N}^n$ (i.e. a binary relation on $\mathbb{N}^n$ that is reflexive, antisymmetric, transitive and total), we say that it is a **monomial order** if, for all $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{N}^n$ we have:

- $(0, \dots, 0) \prec \boldsymbol{a}$
- $\boldsymbol{a} \prec \boldsymbol{b}$ implies $\boldsymbol{a} + \boldsymbol{c} \prec \boldsymbol{b} + \boldsymbol{c}$

An important example is the lexicographical order, in which $\boldsymbol{a} <_{lex} \boldsymbol{b}$ if the leftmost nonzero entry of $\boldsymbol{b} - \boldsymbol{a}$ is positive. For the lexicographical we can also change the order of the variables using a permutation.

We can define the initial of a polynomial f with respect to the monomial order $\prec$ as the $\prec$-largest monomial between the one appearing with non-zero coefficent in f. Given an ideal I of a polynomial ring we can define also the *initial ideal* as:

$$in_{\prec}(I) = \langle in_{\prec}(f) \ : f \in I \setminus \{0\} \rangle$$

Proposition

For any field k and monomial order $\prec$, given an ideal I there exists a finite subset $\mathcal{G}$ such that:

$$in_{\prec}(I) = \langle in_{\prec}(f) : f \in \mathcal{G} \rangle$$

In this case $\mathcal{G}$ is called a ***Groebner basis*** *for I with respect to $\prec$.*

Reduced Groebner basis

It is obvious from the previous definition that the Groebner Basis is not unique, but we can achieve this with the following requirements:

Definition

A Groebner basis $\mathcal{G}$ for the ideal I with respect to $\prec$ is *reduced* if the following holds:

- Each polynomial of $\mathcal{G}$ is monic.
- For each $f, g \in \mathcal{G}$ we have that $in_{\prec}(h)$ does not divide any monomial of g.

It is possible to prove that any ideal has a unique reduced groebner basis.

Calculation of Groebner Basis I

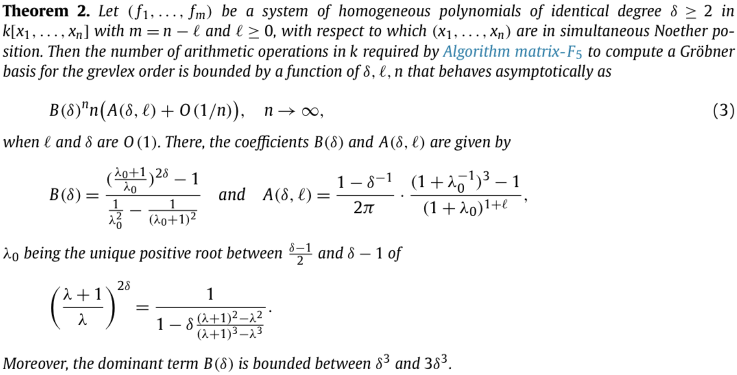
The most known algorithm for computing Groebner basis is the **Bucheberg algorithm**, it starts from a set F of generators for the ideal, then:

1. Define $G := F$
2. Insert all the pairs of different elements of G in the set P
3. Until the set P is empty take an element in it and compute the normal form h of its s-polynomial with respect to G. If $h \neq 0$ then:
   1. Add to P the pairs (h, g) for all $g \in G$.
   2. Add h to G.

Calculation of Groebner Basis II

As you can see from the the complexity of the algorithm is clearly at least exponential, in fact computing Groebner basis is a very difficult task, even for easy ideals. At today state of the art the most efficients algorithms are the Faugère F4 and F5, that are implemented in:

- SageMath implements both of them
- MAGMA implements F4
- Maple implements F4
- SINGULAR implements F5
- Faugère's own implementation of F4 can be found on [Fau]



Theorem 2. *Let* $(f_1, \ldots, f_m)$ *be a system of homogeneous polynomials of identical degree* $\delta \geq 2$ *in* $k[x_1, \ldots, x_n]$ *with* $m = n - \ell$ *and* $\ell \geq 0$, *with respect to which* $(x_1, \ldots, x_n)$ *are in simultaneous Noether position. Then the number of arithmetic operations in k required by Algorithm matrix-F*$_5$ *to compute a Gröbner basis for the grevlex order is bounded by a function of* δ, ℓ, n *that behaves asymptotically as*

$$B(\delta)^n n\big(A(\delta, \ell) + O(1/n)\big), \quad n \to \infty, \tag{3}$$

when ℓ *and* δ *are* $O(1)$. *There, the coefficients* $B(\delta)$ *and* $A(\delta, \ell)$ *are given by*

$$B(\delta) = \frac{(\frac{\lambda_0+1}{\lambda_0})^{2\delta} - 1}{\frac{1}{\lambda_0^2} - \frac{1}{(\lambda_0+1)^2}} \quad \text{and} \quad A(\delta, \ell) = \frac{1 - \delta^{-1}}{2\pi} \cdot \frac{(1 + \lambda_0^{-1})^3 - 1}{(1 + \lambda_0)^{1+\ell}},$$

λ_0 *being the unique positive root between* $\frac{\delta-1}{2}$ *and* $\delta - 1$ *of*

$$\left(\frac{\lambda + 1}{\lambda}\right)^{2\delta} = \frac{1}{1 - \delta\frac{(\lambda+1)^2 - \lambda^2}{(\lambda+1)^3 - \lambda^3}}.$$

Moreover, the dominant term $B(\delta)$ *is bounded between* δ^3 *and* $3\delta^3$.

Figure: Complexity of the F5 algorithm from [BFS15]

Elimination Theorem

Theorem (Elimination theorem)

Set $R = \mathbb{F}[x_1, ..., x_n]$*, and use the order* $<_{lex}$ *with* $x_1 <_{lex} x_2 <_{lex} ... <_{lex} x_n$*.*
Let $I \subset R$ *be an ideal and* G *a Groebner basis of* I *with respect to* $<_{lex}$*. Then* $G \cap \mathbb{F}[x_1, ..., x_l]$ *is a Groebner basis of* $I \cap \mathbb{F}[x_1, ..., x_l]$*.*

Error locator polynomial

Error locator polynomial

If we have $j_l, 1 \leq l \leq t$ as the error positions for the received word we would like to find the *error locator polynomial*, that is a polynomial having as roots the error locations ξ^{j_l}:

$$L(z) := \prod_{l=1}^{t} (z - \xi^{j_l}) \tag{6}$$

Observe that a polynomial of this kind should be in the syndrome variety when considered the evaluation of the known syndromes and intersected with $\mathbb{F}_q[z_1]$.

Maybe we can use **Groebner Basis**?

Error locator polynomial

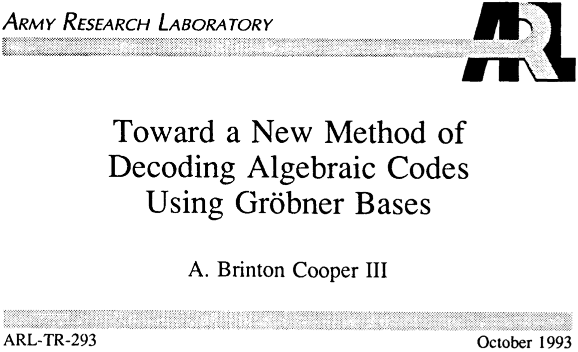
If we have $j_l, 1 \leq l \leq t$ as the error positions for the received word we would like to find the *error locator polynomial*, that is a polynomial having as roots the error locations ξ^{j_l}:

$$L(z) := \prod_{l=1}^{t}(z - \xi^{j_l}) \tag{6}$$

Observe that a polynomial of this kind should be in the syndrome variety when considered the evaluation of the known syndromes and intersected with $\mathbb{F}_q[z_1]$.

Maybe we can use **Groebner Basis**?

Cooper's Philosophy I



ARMY RESEARCH LABORATORY

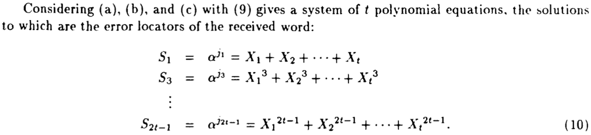
Toward a New Method of Decoding Algebraic Codes Using Gröbner Bases

A. Brinton Cooper III

ARL-TR-293 October 1993

Figure: That's actually the idea of Cooper in the article [Coo92]

Cooper's Philosophy II

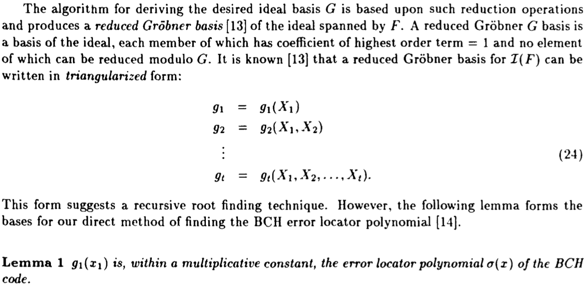


Considering (a), (b), and (c) with (9) gives a system of t polynomial equations, the solutions to which are the error locators of the received word:

$$\begin{aligned} S_1 &= \alpha^{j_1} = X_1 + X_2 + \cdots + X_t \\ S_3 &= \alpha^{j_3} = {X_1}^3 + {X_2}^3 + \cdots + {X_t}^3 \\ &\vdots \\ S_{2t-1} &= \alpha^{j_{2t-1}} = {X_1}^{2t-1} + {X_2}^{2t-1} + \cdots + {X_t}^{2t-1}. \end{aligned} \tag{10}$$

Figure: These are the polynomials f_u in $\mathbb{F}_2$ with the assumptions that exactly t errors occurred and using χ_u to remove equations

Cooper's Philosophy III



The algorithm for deriving the desired ideal basis G is based upon such reduction operations and produces a *reduced Gröbner basis* [13] of the ideal spanned by F. A reduced Gröbner G basis is a basis of the ideal, each member of which has coefficient of highest order term = 1 and no element of which can be reduced modulo G. It is known [13] that a reduced Gröbner basis for $\mathcal{I}(F)$ can be written in *triangularized* form:

$$\begin{array}{rcl} g_1 & = & g_1(X_1) \\ g_2 & = & g_2(X_1, X_2) \\ \vdots & & \\ g_t & = & g_t(X_1, X_2, \ldots, X_t). \end{array} \tag{24}$$

This form suggests a recursive root finding technique. However, the following lemma forms the bases for our direct method of finding the BCH error locator polynomial [14].

Lemma 1 *$g_1(x_1)$ is, within a multiplicative constant, the error locator polynomial $\sigma(x)$ of the BCH code.*

Figure: Here we are using elimination theorem (8) and that its roots are the error locations

General error locator polynomial

Definition

Let $L_{\mathcal{C}}$ be a polynomial in $\mathbb{F}_q[x_1, \dots, x_r, z]$.Then $L_{\mathcal{C}}$ is a *general error locator polynomial* of $\mathcal{C}$ if

1. $L_{\mathcal{C}} = z^t + a_{t-1}z^{t-1} + \dots + a_0$, with $a_j \in \mathbb{F}_q[x_1, \dots, x_r]$ for all j
2. Given the syndromes $s_1, \dots, s_r \in \mathbb{F}_q$, corresponding to an error of weight μ and error locations $\{k_1, \dots, k_\mu\}$, if we evaluate the x_i variables with s_i, then the roots of $L_{\mathcal{C}}(s_1, \dots, s_r, z)$ are exactly $\{\xi^{k_1}, \dots, \xi^{k_\mu}, 0, \dots, 0\}$, i.e.

$$L_{\mathcal{C}}(s_1, \dots, s_r, z) = z^{n-\mu} \prod_{l=1}^{\mu} (z - \xi^{k_l}) \qquad (7)$$

Finding the general error locator polynomial

Goal

Use the CHRT-syndrome ideal to find the *general error locator polynomial* associated to the code $\mathcal{C}$ using the Elimination Theorem

The problem is that now the variety contains too many points, we need to remove some of them, called also spurious.

Spurious points

In the article [OS05] they observed that such points are of the type:

$$(\xi^{k_1}, ..., \xi^{k_\mu}, \zeta, \zeta, 0, ..., 0, \hat{y}_1, ..., \hat{y}_\mu, Y, -Y, y_1, ..., y_{t-(\mu+2)}) \qquad (8)$$

Solution

We can solve this adding the polynomials:

$$p_{i,j} := z_i z_j \frac{z_i^n - z_j^n}{z_i - z_j}$$

Define so $F'_{\mathcal{C}}$ as the union of $F_{\mathcal{C}}$ and $p_{i,j}$ for $1 \leq i < j \leq t$.

Spurious points

In the article [OS05] they observed that such points are of the type:

$$(\xi^{k_1}, ..., \xi^{k_\mu}, \zeta, \zeta, 0, ..., 0, \hat{y}_1, ..., \hat{y}_\mu, Y, -Y, y_1, ..., y_{t-(\mu+2)}) \tag{8}$$

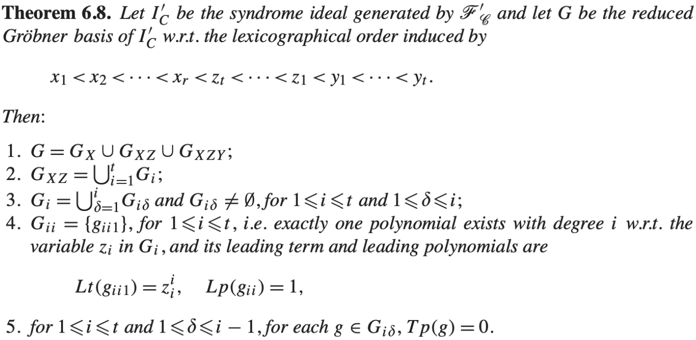
Solution

We can solve this adding the polynomials:

$$p_{i,j} := z_i z_j \frac{z_i^n - z_j^n}{z_i - z_j}$$

Define so $F'_{\mathcal{C}}$ as the union of $F_{\mathcal{C}}$ and $p_{i,j}$ for $1 \leq i < j \leq t$.

A particular structure for Groebner basis



Theorem 6.8. *Let I'_C be the syndrome ideal generated by $\mathscr{F}'_{\mathscr{C}}$ and let G be the reduced Gröbner basis of I'_C w.r.t. the lexicographical order induced by*

$$x_1 < x_2 < \cdots < x_r < z_t < \cdots < z_1 < y_1 < \cdots < y_t.$$

Then:

1. $G = G_X \cup G_{XZ} \cup G_{XZY}$;
2. $G_{XZ} = \bigcup_{i=1}^{t} G_i$;
3. $G_i = \bigcup_{\delta=1}^{i} G_{i\delta}$ *and* $G_{i\delta} \neq \emptyset$, *for* $1 \leqslant i \leqslant t$ *and* $1 \leqslant \delta \leqslant i$;
4. $G_{ii} = \{g_{ii1}\}$, *for* $1 \leqslant i \leqslant t$, *i.e. exactly one polynomial exists with degree* i *w.r.t. the variable* z_i *in* G_i, *and its leading term and leading polynomials are*

$$Lt(g_{ii1}) = z_i^i, \quad Lp(g_{ii}) = 1,$$

5. *for* $1 \leqslant i \leqslant t$ *and* $1 \leqslant \delta \leqslant i - 1$, *for each* $g \in G_{i\delta}$, $Tp(g) = 0$.

Figure: From the article [OS05]

Where we have that:

1. G_X is the Groebner basis intersected with $\mathbb{F}_q[x_1, ..., x_r]$
2. $G_i = G \cap \mathbb{F}_q[x_1, ..., x_r, z_t, ..., z_i]$
3. $G_{i\delta} = \{g \in G_i \setminus G_{i+1} : \deg_{z_i}(g) = \delta\}$
4. $g_{ii1} = z_i^i + \sum_{l=0}^{i-1} a_l z_i^l$ for $a_l \in \mathbb{F}_q[x_1, ..., x_r]$

Remark

It is possible to see that for g_{tt1} are equivalent:

- There are exactly $\mu \leq t$ errors
- $a_l(\boldsymbol{s}) = 0$ for $l < t - \mu$

Main result

Theorem (Theo 6.9 [OS05])

Each cyclic code $\mathcal{C}$ admits a general error locator polynomial $L_{\mathcal{C}}$, that is also an element of the Groebner basis of the ideal generated by:

$$F'_{\mathcal{C}} = \{f_u, \chi_u, h_l, \lambda_l, p_{i,j} \text{ for } 1 \leq u \leq r, 1 \leq l \leq t, 1 \leq i < j \leq t\}$$

with the lexicographical order induced by

$$x_1 < x_2 < \ldots < x_r < z_t < \ldots < z_1 < y_1 < \ldots < y_t$$

Proof.

It is enough to use theorem in figure 43, in particular we have to take the polynomial

$$g_{tt1}(x_1, ..., x_r, z_t),$$

that is unique and with the required properties of degrees, ring of definition and leading term equal to 1.

Proof.

We need only to prove that the roots are exactly the error locations. This is proven in Lemma 6.4 of [OS05].

Proof.

In particular given the known syndromes we can define $I_{\mathcal{C}}^{\boldsymbol{s}} := I'_{\mathcal{C}} \cap \langle x_{i_u} = s_{i_u} \rangle_{1 \leq u \leq r}$, such that $\mathcal{V}(I_{\mathcal{C}}^{\boldsymbol{s}})$ are the extension of the errors locations and values for the known syndromes.

At this point we have that:

$$\mathcal{V}(g_{tt1}) \supseteq \mathcal{V}(G_t) \overset{\text{Elim}}{=} \mathcal{V}(I_{\mathcal{C}}^{\boldsymbol{s}} \cap \mathbb{F}_q[z_t]) \supseteq \pi(\mathcal{V}(I_{\mathcal{C}}^{\boldsymbol{s}})) = \{0, \xi^{k_1}, \dots, \xi^{k_t}\}$$

And using the remark 3 we can end the proof. ☐

Matroids

Generalization of independence

The concept of **matroid** generalize the ideas of linear independence and of *cycle free* in graph theory.

The three key properties that we want to generalize are:

1. the empty set is linear independent
2. a subset of a set of linear independent vectors is again linear independent
3. Given two sets of linear independent vectors, one greater than the other, is possible to extend the smaller one with a vector of the other set

Definition

Definition

A **matroid** is a pair $(E, \mathcal{I})$ where E is a finite set and $\mathcal{I}$ a collection of subset of E such that

1. $\emptyset \in \mathcal{I}$
2. If $I \in \mathcal{I}$ and $S \subset I$ then $S \in \mathcal{I}$
3. If $I, J \in \mathcal{I}$ with $|I| < |J|$ then there exists $j \in J \setminus I$ such that $I \cup \{j\} \in \mathcal{I}$

Associated objects

For any matroid $M := (E, \mathcal{I})$ we can define the following objects:

dependent sets	$\mathcal{D} = \{D \subseteq E : D \notin \mathcal{I}\}$
circuits	$\mathcal{C} = \{C \subseteq E : C \notin \mathcal{I}, \forall c \in C : C \backslash \{c\} \in \mathcal{I}\}$
rank function	$r(J) = \max \{\lvert J' \rvert : J' \subseteq J, J' \in \mathcal{I}\}$
bases	$\mathcal{B} = \{B \subseteq E : r(B) = \lvert B \rvert = r(E)\}$
flats	$\mathcal{F} = \{F \subseteq E : \forall e \in E \backslash F : r(F \cup \{e\}) > r(F)\}$

Any of these can be used to define the matroid uniquely.

Matroids associated to matrices

Consider a $k \times n$ matrix G in a finite field $\mathbb{F}$, this matrix define a code $\mathcal{C}$ when seen as generator matrix.
We can associate a matroid $M_G := (E, \mathcal{I}_G)$ to G defined as:

- $E = \{1, ..., n\}$, the set indexing the columns of G
- $\mathcal{I}_G$ contains the subsets I such that the columns $\{G_i\}_{i \in I}$ are linearly independent

Matroids associated to codes

Proposition

If G_1, G_2 *two generator matrix of the same* $[n, k]$*-code* $\mathcal{C}$ *then*

$$M_{G_1} = M_{G_2}$$

So we can define the matroid $M_{\mathcal{C}}$ associated to the linear code $\mathcal{C}$ as M_G for any G generator matrix.

Definition

Let $M_1 = (E_1, \mathcal{I}_1)$ and $M_2 = (E_2, \mathcal{I}_2)$ be matroids. A map $\phi : E_1 \to E_2$ is called a *morphism* of matroids if I dependent in M_1 implies $\phi(I)$ dependent in M_2.

$\phi : M_1 \to M_2$ is an *isomorphism* if it is invertible and $I \in \mathcal{I}_1$ if and only if $\phi(I) \in \mathcal{I}_2$

Definition

Let $M = (E, \mathcal{I})$ be a matroid, the we can define the *dual matroid* $M^* = (E, \mathcal{I}^*)$ as $\mathcal{I}^* := \{\, I \subseteq E \mid \exists B \in \mathcal{B}. I \subset E \setminus B \}$.

Analogies between duals I

Theorem

Let $\mathcal{C}$ be a linear code, then we have that

$$(M_{\mathcal{C}})^* \simeq M_{\mathcal{C}^\perp}$$

Proof

The isomorphism map is the identity. Now consider an independent subset I of the dual matroid, without loss of generality we can assume that I is contained in the complement of the basis $\{1, ..., k\}$.

Analogies between duals II

Proof.

Since we have seen that from proposition 6.1 we can choose arbitrarly the generator matrix and assume it to be in systematic form. So we have that:

$$G = (Id_k | R) \text{ and } H = (R^\top | I_{n-k})$$

And so we have that I is trivially indipendent for M_H.

The other implication is analogue, we only have to assume for I to be contained in the basis $\{k+1, ..., n\}$ and use the same idea. ☐

Analogies between duals II

Proof.

Since we have seen that from proposition 6.1 we can choose arbitrarily the generator matrix and assume it to be in systematic form. So we have that:

$$G = (Id_k | R) \text{ and } H = (R^\top | I_{n-k})$$

And so we have that I is trivially indipendent for M_H.

The other implication is analogue, we only have to assume for I to be contained in the basis $\{k+1, ..., n\}$ and use the same idea. ☐

Analogies between MDS and uniform I

Definition

Let n and k be non-negative integers such that $k \leq n$. Let $\mathcal{I}_{n,k} = \{I \subset [n] : |I| \leq k\}$. Then $U_{n,k} = ([n], I_{n,k})$ is a matroid that is called the *uniform matroid* of rank k on n elements.

Fixed the parameters n, k from the Singleton bound we have that $d \leq n - k + 1$, a code is maximum distance separable (MDS) code if it achieve equality.

Proposition

An $[n, k]$-code $\mathcal{C}$ is MDS if and only if the matroid $M_{\mathcal{C}}$ is the uniform matroid

Analogies between MDS and uniform II

To prove the previous proposition it is enough to use the following theorem:

Theorem (Proposition 2.2.5 of [Pel+17])

Let $\mathcal{C}$ be an $[n, k, d]$ code with G as generator matrix and H as parity check matrix. Then are equivalent:

1. *$\mathcal{C}$ is an MDS code,*
2. *every $(n-k)$-tuple of columns of a parity check matrix H are linearly independent,*
3. *every k-tuple of columns of a generator matrix G are linearly independent.*

Analogies between MDS and uniform III

In the previous theorem the implication $\mathbf{1} \leftrightarrow \mathbf{2}$ is a classical result from coding theory, while the implication *2 if and only if 3* can be proved using matroids and theorem 14.
Infact the thesis becomes:

$$M_G \text{ uniform} \iff (M_G)^* \text{ uniform}$$

Bibliography

Bibliography I

[BFS15] M. Bardet, J.-C. Faugère, and B. Salvy. “On the complexity of the F5 Gröbner basis algorithm”. In: *Journal of Symbolic Computation* 70 (2015), pp. 49–70. DOI: `https://doi.org/10.1016/j.jsc.2014.09.025`.

[Che+94a] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong. “Algebraic decoding of cyclic codes: A polynomial ideal point of view”. English. In: *Finite fields: theory, applications and algorithms. 2nd international conference, August 17-21, 1993, Las Vegas, NV, USA*. Providence, RI: American Mathematical Society, 1994, pp. 15–22.

Bibliography II

[Che+94b] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong. "General principles for the algebraic decoding of cyclic codes". English. In: *IEEE Trans. Inf. Theory* 40.5 (1994), pp. 1661–1663. DOI: 10.1109/18.333886.

[Che+94c] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong. "Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance". English. In: *IEEE Trans. Inf. Theory* 40.5 (1994), pp. 1654–1661. DOI: 10.1109/18.333885.

Bibliography III

[Coo92] I. Cooper A. B. “Toward a New Method of Decoding Algebraic Codes Using Groebner Bases”. In: *ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD* (1992). URL: https://apps.dtic.mil/sti/citations/ADA273089.

[Fau] J.-C. Faugère. *FGb*. https://www-polsys.lip6.fr/~jcf/FGb/index.html. Accessed: 2022-05-09.

Bibliography IV

[Fau99] J.-C. Faugére. "A new efficient algorithm for computing Gröbner bases (F4)". In: *Journal of Pure and Applied Algebra* 139.1 (1999), pp. 61–88. DOI: `https://doi.org/10.1016/S0022-4049(99)00005-5`.

[MS21] M. Michałek and B. Sturmfels. *Invitation to nonlinear algebra*. English. Vol. 211. Grad. Stud. Math. Providence, RI: American Mathematical Society (AMS), 2021.

Bibliography V

[OS05] E. Orsini and M. Sala. “Correcting errors and erasures via the syndrome variety”. In: *Journal of Pure and Applied Algebra* 200.1 (2005), pp. 191–226. DOI: `https://doi.org/10.1016/j.jpaa.2004.12.027`.

[Pel+17] R. Pellikaan, X.-W. Wu, S. Bulygin, and R. Jurrius. *Codes, Cryptology and Curves with Computer Algebra*. Cambridge University Press, 2017. DOI: `10.1017/9780511982170`.

[Sal+09] M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso, eds. *Gröbner bases, coding, and cryptography*. English. Berlin: Springer, 2009. DOI: `10.1007/978-3-540-93806-4`.