Coding Theory and Applications

Cyclic Codes

Enes Pasalic
University of Primorska
Koper, 2013

Contents

Chapter 1

Preface

This book has been written as lecture notes for students who need a grasp of the basic principles of cyclic codes.

The scope and level of the lecture notes are considered suitable for undergraduate students of Mathematical Sciences at the Faculty of Mathematics, Natural Sciences and Information Technologies at the University of Primorska.

It is not possible to cover here in detail every aspect of cyclic codes, but I hope to provide the reader with an insight into the essence of the cyclic codes.

Enes Pasalic
enes.pasalic@upr.si

Chapter 2

Basics on finite fields and Introduction to cyclic codes

Contents of the chapter:

- Finite fields
- Cyclic codes

Groups - reminder

- *Group* is a set G together with an operation "$\circ$" satisfying:

  1. $\forall a, b \in G : a \circ b \in G$ Algebraic closure
  2. $\forall a, b, c \in G : a \circ (b \circ c) = (a \circ b) \circ c$ Associativity
  3. $\exists! e \in G : \forall a \in G : a \circ e = e \circ a = a$ e is identity element
  4. $\forall a \in G, \exists a^{-1} \in G : a \circ a^{-1} = a^{-1} \circ a = e$ Inverse element

- $(G, \circ)$ is called Abelian if for all $a, b \in G$, $a \circ b = b \circ a$

Example of Groups

Example

- $(\mathbb{Z}, +)$ is a group under usual integer addition. We check,

$$\forall a \in \mathbb{Z}, a + 0 = a; \quad a + (-a) = 0$$

- $(\mathbb{Z}, \cdot)$ is not a group as,

$$3^{-1} =? \text{ i.e. } \quad 3 \cdot x = 1 \quad \text{has no solution in } \mathbb{Z}$$

Example

- $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus 0 = \{1, 2, \ldots, p-1\}$ is a group under multiplication $\pmod p$ iff p is prime.
- For example, $(\mathbb{Z}_5^*, \cdot \pmod 5)$ is a group since,

$$1^{-1} = 1; \; 2^{-1} = 3; \; 3^{-1} = 2; \; 4^{-1} = 4;$$

Example of Groups

- Used bitwise addition to vector space into cosets (st. array)

Example

Given a (4,2,2) code C generated by,

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

we could split the space $V_2(4)$ into 4 cosets,

$$\begin{aligned} &C \\ &(0001) + C \\ &(0010) + C \\ &(0100) + C \end{aligned}$$

Example of Groups

- What about other operations on vectors in $V_2(4)$?
- How can we multiply vectors (10) and (11) from $V_2(2)$?
- We may try to perform the operations in $Z_4 = \{0, 1, 2, 3\}$ by associating

$$(00) = 0;\ (01) = 1;\ (10) = 2; (11) = 3$$

- But this does not work $2^{-1} =?$
- We need another structure to do this !

More complex structures-Rings

- We need two algebraic operations "+" and "·" on a set.

Definition
A set R together with "+" and "·" is a ring if,

1. $(R, +)$ is abelian group with $\mathbf{0}$ as "additive" identity.
2. R is closed under "·" and $\mathbf{1} \neq \mathbf{0} \in R$, $\mathbf{1}$ is multiplicative identity
3. For all $a, b, c \in R$ we have $a \cdot (b + c) = a \cdot b + a \cdot c$. (Distributivity)

Examples of Rings

- An important example of a ring is a polynomial ring over $\mathbb{Z}_2 = \{0, 1\}$
- Its elements are formal polynomials of the form,

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + \cdots + a_k x^k; \quad a_i \in \mathbb{Z}_2$$

- The ring formed by all formal polynomials with binary coefficients is called a binary ring $\mathbb{Z}_2[x]$
- This concept can be generalized to several indeterminates $x_1, \ldots, x_n$ - a ring $\mathbb{Z}_2[x_1, \ldots, x_n]$(Boolean functions).

Examples of Rings II

- The operations $+$ and $\cdot$ are usual polynomial addition and multiplications but coefficients are still in $\mathbb{Z}_2$.
- This means that addition of $f(x) = x + x^2$ and $g(x) = 1 + x + x^2$ gives

$$f(x) + g(x) = 1, \text{ for } x + x = 0 \text{ in } \mathbb{Z}_2[x]$$

- Possible representation : (011)+(111)=(100) as we already had.

Examples of Rings III

Problem is the multiplication :

$$f(x) \cdot g(x) = x + x^4 \leftrightarrow (01001)$$

Cannot represent with 3 bits as addition !

Need a closed structure to be able to multiply codewords.

In other words we need a **field**.

Finite Fields - definition

Definition

Let $(R, +)$ be abelian group. If $(R \setminus \mathbf{0}, \cdot)$ is a group then $(R, +, \cdot)$ is called a *field*.

Example

E.g. $(\mathbb{Z}_p, +, \cdot)$ (mod p) is a field for prime p

Facts

To construct fields of nonprime order - need an irreducible polynomial $f(x) \in \mathbb{F}_2[x]$, i.e. $f(x) = g(x)h(x)$ implies that either g or h is a constant polynomial.

Irreducible polynomials - example

Example

The polynomial $f(x) = x^3 + x + 1$ is irreducible over $\mathbb{F}_2$ whereas,

$$r(x) = x^3 + x^2 + x + 1 = (x^2 + 1)(x + 1)$$

is reducible.

Alternatively, there is no root of $f(x) = x^3 + x + 1$ in $\mathbb{Z}_2$

$$f(0) = 1 \quad f(1) = 1.$$

For $r(x)$ we have $r(1) = 0$.

Construction of finite fields of nonprime order

Definition

The degree of $f(x) = \sum_{i=0}^{n} a_i x^i$ is the largest i for which $a_i \neq 0$.

Example

The degree of $f(x) = x^3 + x + 1$ is 3, and degree of $g(x) = x + 1$ is 1.

IMPORTANT: We can make our multiplication closed if we reduce modulo some polynomial of degree n

Reduction modulo $x^2 + x + 1$ - example

Example

Previously problem was that multiplication was not "closed". We use 2 bit representation of polynomials of degree $\leqslant 1$ but,

$$x(1 + x) = x + x^2 \leftrightarrow (011)$$

If we compute modulo $x^2 + x + 1$ then

$$x(1 + x) = x + x^2 = 1 \pmod{x^2 + x + 1}$$

Simply use $x^2 + x + 1 = 0$ as we use $5 \pmod 5 = 0$.

Polynomial reduction

In above example reduction was easy. How to do it in general ?

Example

Want to compute $x^4 + x^2 + x + 1 \pmod{x^2 + x + 1}$. Need to perform "long division"

$$
\begin{array}{cccccc}
 & & x^2 & +x & +1 & \\
\cline{2-6}
x^2+x+1) & x^4 & +x^2 & +x & +1 & \\
 & x^4 & +x^3 & +x^2 & & \\
\cline{2-6}
 & & x^3 & +x & +1 & \\
 & & x^3 & +x^2 & +x & \\
\cline{2-6}
 & & & x^2 & +1 & \\
 & & & x^2 & x+1 & \\
\cline{2-6}
 & & & & x &
\end{array}
$$

Polynomial reduction

Our division says that,

$$x^4 + x^2 + x + 1 = (x^2 + x + 1)(x^2 + x + 1) + x$$

which means

$$x^4 + x^2 + x + 1 = x \pmod{x^2 + x + 1}.$$

What about multiplicative inverses of elements $\{1, x, (1 + x)\}$?

We already showed that $x(1 + x) = 1 \pmod{x^2 + x + 1}$. Hence all the elements have inverses ! We can finally do the computation $(01) \cdot (11) = (10)$!

Construction of finite fields of nonprime order II

- One can prove that using an irreducible f of degree n we construct a field of 2^n elements,

  $\mathbb{Z}_2[x]/\pmod{f(x)} = \{$ all polynomials of degree less than $n\}$.

- The operations on polynomials $p(x), q(x) \in \mathbb{Z}_2[x]/\pmod{f(x)}$ are defined by,

$$\begin{aligned} p(x) + q(x) &= p(x) + q(x) \pmod{f(x)} \\ p(x) \cdot q(x) &= p(x) \cdot q(x) \pmod{f(x)} \end{aligned}$$

Examples of construction

- Construction of $\mathbb{F}_{2^3}$ using $f(x) = x^3 + x + 1$.

$$\begin{array}{lll} x^0 & = 1 & (100) \\ x^1 & = x & (010) \\ x^2 & = x^2 & (001) \\ x^3 & = x + 1 \pmod{x^3 + x + 1} & (110) \\ x^4 & = x^2 + x \pmod{x^3 + x + 1} & (011) \\ x^5 & = x^2 + x + 1 \pmod{x^3 + x + 1} & (111) \\ x^6 & = x^2 + 1 \pmod{x^3 + x + 1} & (101) \\ x^7 & = 1 \pmod{x^3 + x + 1} & (100) \end{array}$$

- Verify that for e.g. $x^5 = (x^2 + 1)(x^3 + x + 1) + x^2 + x + 1$.

Connection to coding - representation

- The polynomial $p(x) = x^2 + x + 1$ is irreducible (and primitive) over $\mathbb{F}_2$. Hence

$$\mathbb{F}_{2^2} = \{0, x^0, x^1, x^2 = 1 + x\} = \{0, 1, \alpha, \beta\}$$

- The elements can be represented as binary tuples $\{(00), (10), (01), (11)\}$
- Connection to coding:

$$\mathbf{c} = (\alpha, 1, \beta) \rightarrow (01, 10, 11)$$

Sometimes good codes (Reed-Solomon) are found using large alphabets $\alpha, \beta \in \mathbb{F}_{2^k}$.

An example of extension fields

- To construct $\mathbb{F}_{2^4}$ we consider the polynomial

$$p(y) = a_2 y^2 + a_1 y + a_0 \in \mathbb{F}_{2^2}[y]$$

where $a_i \in \mathbb{F}_{2^2}$.

- Note that $p(y) = y^2 + y + 1$ is reducible over $\mathbb{F}_{2^2}$ as $p(y) = (y + \alpha)(y + \beta)$ for $\alpha = x + 1$ and $\beta = x$.
- It can be checked that $y^2 + xy + 1$ is irreducible (exercise).
- Then our extension field is defined as a set of polynomials $\{a_1 y + a_0; a_0, a_1 \in \mathbb{F}_{2^2}\}$ with reduction $y^2 = xy + 1$.
- The elements are (a_1, a_0) - 4 bit binary vectors.

Extension fields

- We may construct the finite field $\mathbb{F}_{2^n}$ by using an irreducible polynomial of degree n over $\mathbb{F}_2$ (a finite extension of $\mathbb{F}_2$).
- The "same" field can be constructed using a tower of fields.

- E.g. let $k|n$ and $n/k = s$. Then we may first construct $\mathbb{F}_{2^k}$ using an irreducible polynomial $f(x)$ over $\mathbb{F}_2$ of degree k. This field is $\mathbb{F}_2[x]/\pmod{f(x)} = \mathbb{F}_{2^k}$.
- Now let $g(y)$ be an irreducible polynomial over $\mathbb{F}_{2^k}$ of degree s. Then $\mathbb{F}_{2^k}[y]/\pmod{g(y)} = \mathbb{F}_{2^n}$.

Subfields of finite fields

- Suppose $GF(q) \subset GF(Q)$ have characteristic p. Then

$$q = p^m, Q = p^M, m \leqslant M.$$

$GF(Q)$ is a vector space over $GF(q)$ of dimension n.

$$Q = q^n \Rightarrow p^M = (p^m)^n = p^{mn} \Rightarrow M = nm \Rightarrow m|M.$$

- The exponent of the subfield divides the exponent of the extension field.

Subfields of finite fields -example

Example

Subfields of GF(2^{35}) are GF(2^7), GF(2^5), GF(2).

GF(2^{19}) is not contained in GF(2^{35}) since 19 is not a divisor of 35.

Smallest subfield that contains both GF(2^{19}) and GF(2^{35}) is

$$GF(2^{lcm(19,35)}) = GF(2^{19 \cdot 35}) = GF(2^{665}).$$

Cyclic codes - preliminaries

One of the most important classes of linear codes.

Importance of this class due to:

- Generic construction
- Many other linear codes are derived from cyclic codes
- Efficient implementation using shift registers (coding/decoding)

Asymptotically not good but gives a rise to BCH codes, Reed-Solomon codes etc.

Cyclic codes - big picture

Need additional structure for codewords.

Would like to have the property that if,

$$(1100) \in C \quad \text{then} \quad (0110), (0011), (1001) \in C$$

Has to introduce the concept of **cyclic subspace**.

Structure, efficient encoding/decoding but a bit of group theory needed.

Cyclic codes - definition

Definition

A subspace of $V_n(F)$ is a cyclic subspace if

$$(a_1 a_2 \dots a_n) \in S \Leftrightarrow (a_n a_1 a_2 \dots a_{n-1}) \in S$$

In words - S is a subspace and for each $\mathbf{a} \in S$ every cyclic shift of $\mathbf{a}$ is also in S.

Clearly $\mathbf{0} \in S$ and S is closed under addition.

Cyclic subspace - example

Example

- $S = \{(0000), (1111)\} \subset V_4(Z_2)$ is a cyclic subspace.
- But, e.g. S given below is not a cyclic subspace

$$S = \{(0000), (1001), (1100), (0110), (0011), (0, 111), (1011), (1101)\}$$

Constructing cyclic subspaces

A naive approach: take a vector of certain weight and all its cyclic shifts; and add some more vectors so that S is closed under addition. Does it work ?

Example Take (1100) so we must take

$$(1100) \to (0110) \to (0011) \to (1001)$$

Of course we must take **0** and forced to take,

$$(1010), (0101), (1111).$$

It worked here but not always (exercise - find counterexamples)

Fundamental questions on cyclic codes

The most important questions to be answered are :

1. How can cyclic subspaces be constructed ?
2. For a given k can a k-dimensional subspace be constructed ?
3. How many cyclic subspaces $V_n(\mathbb{F})$ contain ?
4. Which vectors have the property that the vector and its cyclic shifts will generate all of S ?

Example of cyclic subspace

Example

Consider 4-dimensional subspace $S \subset V_6(Z_2)$ generated by,

$$\mathbf{v}_1 = (111000), \mathbf{v}_2 = (011100), \mathbf{v}_3 = (001110), \mathbf{v}_4 = (000111)$$

Why the cyclic shifts of $\mathbf{v}_1 + \mathbf{v}_2 = (100100)$ are in the subspace ?

For instance (0100100) is in the subspace as $(0100100) = \mathbf{v}_2 + \mathbf{v}_3$

. Thus $\mathbf{v}_1$ and its 3 cyclic shifts generate S.

Rings of polynomials

- Need to introduce the ring of polynomials over $\mathbb{F}$ modulo $f(x)$, where $f(x)$ is not irreducible, $R(+,*)$!!
- Same as for fields - form equivalence classes (cosets) using mod $f(x)$ relation.

$$[g(x)] = \{h(x) \in \mathbb{F}[x] : h(x) \equiv g(x) \pmod{f(x)}\}.$$

- For instance, if $f(x) = x^3 + 1$ then

$$[1+x^2]*[1+x+x^2] = [1+x+x^3+x^4] = [x+x^4] = [x(1+x^3)] = 0.$$

Need the concept of ideals to study cyclic subspaces

Ideals

Definition
Let $R(+,*)$ be a ring. A nonempty subset I of R is called an **ideal** if:

(i) $(I,+)$ is a group, and

(ii) $i * r \in I$ for all $i \in I$ and $r \in R$

- Goal: Establish 1-1 correspondence between ideals in $\mathbb{F}[x]/x^n - 1$ and cyclic subspaces in $V_n(\mathbb{F})$

Constructing ideals: Take any nonzero $g \in R$ and form the set, $I = \{g * r : r \in R\}$. Then I is an ideal - an ideal generated by g.

Constructing ideals - example

Example

Let $g(x) = x + 1$ and we want to construct an ideal $\pmod{x^3 + 1}$.

$1 \cdot x + 1$	$x + 1$	(110)
$x \cdot x + 1$	$x + x^2$	(011)
$(1 + x)(1 + x)$	$1 + x^2$	(101)
$x^2(1 + x)$	$x^2 + x^3 = x^2 + 1$	(101)
$(1 + x^2)(1 + x)$	$1 + x + x^2 + x^3 = x + x^2$	(011)
$(1 + x + x^2)(1 + x)$	$1 + x^3 = 0$	(000)

We have a cyclic code (ideal) $C = \{(000), (110), (011), (101)\}$

Principal Ideal Rings

– Not all ideals can be constructed in this way ! But some rings are principal ideal rings , i.e.

$$\forall I \subset R \; \exists g \in I : I = \{g * r : r \in R\}.$$

Facts (see the details of the proof in the textbook):

- $F[x]$ is a principal ideal ring
- $F[x]/(f(x)$ is a principal ideal ring

Proof (Sketch): Choose $g \in I$ of min. degree and use the EA,

$[h(x)] = [q(x)g(x) + r(x)] = [q(x)g(x)] + [r(x)]$, $\deg(r) < deg(g)$.

As $[q(x)g(x)] \in I \Rightarrow [h(x) - [q(x)g(x)] = [r(x)] \in I \Rightarrow r(x) = 0$

Principal Ideal Rings - example

Example Consider $Z_2[x]/(x^6+1)$ and the set,

$$I = \{0, 1+x^2+x^4, x+x^3+x^5, 1+x+x^2+x^3+x^4+x^5\}.$$

I is an ideal in R. E.g.

$$x^3 * (1+x^2+x^4) = x^3+x^5+x^7 = x^3+x^5+x.$$

Connection to codes ?

$$I = \begin{cases} 0 \\ 1+x^2+x^4 \\ x+x^3+x^5 \\ 1+x+x^2+x^3+x^4+x^5 \end{cases} \qquad C = \begin{cases} 0 \\ (101010) \\ (010101) \\ (111111) \end{cases}$$

Ideals and cyclic subspaces

Goal: Endow $V_n(\mathbb{F})$ with the operation of multiplication to get a ring. Identify,

$$\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in V_n(\mathbb{F}) \leftrightarrow v(x) = v_0 + v_1 x + \cdots + v_{n-1}x^{n-1}.$$

Multiplication of $\mathbf{v}_1, \mathbf{v}_2 \in V_n(\mathbb{F})$ is defined as

$$v(x) = v_1(x)v_2(x) \bmod x^n - 1.$$

This association identifies $V_n(\mathbb{F})$ and $\mathbb{F}[x]/f(x)$.

Why did we choose $f(x) = x^n - 1$, any polynomial of degree n would do ?

Choice of reduction polynomial

The reason for choosing $f(x) = x^n - 1$ is an efficient implementation.

Consider $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in V_n(\mathbb{F})$. Then,

$$\begin{aligned} v(x) &= v_0 + v_1 x + \cdots + v_{n-1} x^{n-1} \\ xv(x) &= v_0 x + v_1 x^2 + \cdots + v_{n-1} x^n \\ &\overset{x^n=1}{=} v_{n-1} + v_0 x + v_1 x^2 + \cdots + v_{n-2} x^{n-1} \end{aligned}$$

Thus $xv(x) \leftrightarrow (v_{n-1}, v_1, \ldots, v_{n-2})$.

– Now we specify the cyclic subspaces of $V_n(\mathbb{F})$!

Cyclic subspaces - main result

Theorem The linear code C is cyclic **iff** C is an ideal in $\mathbb{F}[x]/(x^n - 1)$.

Proof: If C is an ideal and $(v_0, v_1, \ldots, v_{n-1}) \in C$

$$x(v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}) = (v_{n-1}, v_1, \ldots, v_{n-2}) \in C.$$

Conversely, if $(v_0, v_1, \ldots, v_{n-1}) \in C$ then $(v_{n-1}, v_1, \ldots, v_{n-2}) \in C$.

Then for any $v(x) \in C$ we have $xv(x) \in C$. Therefore,

$$x^2 v(x), x^3 v(x) \ldots \in C \Rightarrow b(x)v(x) \in C \ \forall b(x).$$

Thus, C is an ideal.

Generator polynomial

So far we have established:

- The ring $\mathbb{F}[x]/(x^n-1)$ is a principal ideal ring
- Linear code if cyclic iff C is an ideal in $\mathbb{F}[x]/(x^n-1)$
- Remains: Find generators of ideals in $\mathbb{F}[x]/(x^n-1)$

Theorem Let $I \subset R$ and $g(x)$ a monic polynomial of least degree in I. Then $g(x)$ generates I and $g(x)|f(x) = x^n - 1$

Proof: $g(x)$ obviously generates I. As for the divisibility,

$$\begin{aligned} f(x) &= h(x)g(x) + r(x) \quad \deg(r) < \deg(g) \\ [f(x)] &= [h(x)][g(x)] + [r(x)] \end{aligned}$$

As $[f(x)] = 0$ then $[r(x)] \in I$, and $r(x) = 0$, i.e. $g|f$

Uniqueness of generator polynomial

There is a unique monic polynomial, $g(x)$ of least degree that generates I

Indeed, if $g, h \in I$ and $deg(h) = deg(g)$ then $h(x) = a(x)g(x)$ and $a(x) = 1$.

– $g(x)$ is called generator polynomial

KNOWLEDGE : Generator polynomial of some I must divide $f(x) = x^n - 1$
QUESTION : What about converse? I.e. suppose $h|x^n - 1$ (h monic), form $I = \{a(x)h(x) : a(x) \in R\}$. Is h generator of I ?

Cyclic subspaces - correspondence

The answer is YES, i.e.

If $h|x^n - 1$ and h is monic then h generates $I = \{ah : a \in R\}$

Combining everything so far we have,

There is 1-1 correspondence between cyclic subspaces of $V_n(\mathbb{F})$ and monic polynomials $g(x) \in \mathbb{F}[x]$ that divide $f(x) = x^n - 1$.

Example Consider $V_7(Z_2)$ and $f(x) = x^7 - 1$. Factorization of f over Z_2 is,

$$x^7 - 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1).$$

Example continued

Example cont. Thus, the monic divisors of f are $g_1(x) = 1$, $g_2(x) = x + 1$, $g_3(x) = x^3 + x^2 + 1$, $g_5 = (x+1)(x^3 + x^2 + 1)$ etc.

Exactly 8 cyclic subspaces (follows from factorization). Take now $g_7(x) = (x^3 + x^2 + 1)(x^3 + x + 1)$. What is the dimension of S ?

$$S = \{(0000000), (1111111)\}.$$

Check:
$x(x^3 + x^2 + 1)(x^3 + x + 1) = 1 + x + x^2 + \cdots + x^6 \bmod x^7 - 1$, i.e. (1111111).

For $n = 7$ and $\deg(g) = 6$ we obtain $\dim(S) = 1 = 7 - 6$.
Coincidence ?

Factorization of $x^n - 1$

Only consider here factorization over $GF(2)$. Do we have nontrivial cyclic subspaces for any n ?

There is one unfortunate choice $n = 2^r$ because $1 + x^{2^r} = (1 + x)^{2^r}$ so 1 is the only zero of $f(x)$ with multiplicity 2^r.

–Otherwise, use MAPLE, MAGMA ... and factor, works fine for practical sizes.

- REMARK: Sometimes we have only few possibilities, e.g. $1 + x^{24} = (1 + x)^8(1 + x + x^2)^8$

– Remains to answer the question on the possibility of finding cyclic subspace of a given dimension k !

Cyclic subspaces of dimension k

Theorem Let g be a monic divisor of $x^n - 1$ over $\mathbb{F}$ with $\deg(g) = n - k$. Then g is a generator polynomial for a cyclic subspace of $V_n(F)$ of dimension k

Proof let S be the cyclic subspace of $V_n(\mathbb{F})$ generated by g. CLAIM: $B = \{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ is a basis for S.

B linearly independent: Consider

$$\sum_{i=0}^{k-1} \lambda_i x^i g(x) = 0, \ \lambda_i \in \mathbb{F}$$

As $\deg(g) = n - k$ the only one term which contain x^{n-1} is λ_{k-1}, i.e. $\lambda_{k-1} = 0$. Same for λ_i, $0 \leqslant i \leqslant k - 1$.

Cyclic subspaces of dimension k - proof

B spans S: Let $h(x) \in S$, then $h(x) = a(x)g(x)$.

- W.l.o.g. we may assume that $\deg(a) < k$, otherwise mod reduction. Thus,

$$a(x) = \sum_{i=0}^{k-1} \lambda_i x^i \Rightarrow h(x) = \sum_{i=0}^{k-1} \lambda_i x^i g(x)$$

- So B spans S and $\dim(S) = k$.

Example Want to construct a binary cyclic $(15, 9)$ code. Simply take $g(x) = (1 + x + x^2)(1 + x + x^4)$ and construct the basis $B = \{g(x), xg(x), \ldots, x^8 g(x)\}$

Generator matrix of cyclic codes

Current knowledge: A generator polynomial $g(x)$ of degree $n - k$ generates an (n, k) code C.

- Encoding: Codewords are products $a(x)g(x)$ where $\deg(a) \leqslant k - 1$.
- Message space is a set of polynomials in $\mathbb{F}[x]$ with degree $< k$.

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \\ \vdots \\ x^{k-1} g(x) \end{bmatrix}.$$

Generator matrix - example

From previous example: $g(x) = x^4 + x^2 + x + 1$ generates a cyclic equidistant (7,3) code. The generator matrix is,

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Encoding the message $a = (101)$ means adding the first and the third row of G , i.e. $aG = (1101001)$.

Check,

$$a(x)g(x) = (1 + x^2)g(x) = 1 + x + x^3 + x^6.$$

Summary

– Very interesting class of codes - generic construction and efficient implementation

– Need to analyze decoding algorithms

– Did not discuss the minimum distance of cyclic codes

– Will later see that there is a possibility to design the distance of a cyclic code

– Need to analyze asymptotic behavior

Chapter 3

Cyclic codes

Contents of the chapter:

- Parity check matrix
- Encoding cyclic codes
- Decoding cyclic codes

Cyclic codes - representation

Facts

A codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ is represented as a polynomial in the ring $F[x]/f(x)$

$$c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}$$

In most of the cases we treat $F = \mathbb{Z}_2$ - binary alphabet.

Cyclic codes have additional property:

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow (c_1, \ldots, c_{n-1}, c_0)$$

Doing mathematics is performed cutting polynomials mod $f(x)$.

Cyclic subspaces - reminder

The reason for choosing $f(x) = x^n - 1$ is an efficient implementation.

Consider $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in V_n(\mathbb{F})$. Then,

$$\begin{aligned} v(x) &= v_0 + v_1 x + \cdots + v_{n-1} x^{n-1} \\ xv(x) &= v_0 x + v_1 x^2 + \cdots + v_{n-1} x^n \\ &\overset{x^n=1}{=} v_{n-1} + v_0 x + v_1 x^2 + \cdots + v_{n-2} x^{n-1} \end{aligned}$$

Thus $xv(x) \leftrightarrow (v_{n-1}, v_0, \ldots, v_{n-2})$.

Cyclic subspaces - main result

Theorem

The ring $F[x]/f(x)$ is a principal ideal ring. Means that any ideal in $V_n(F)$ is generated by some element of I.

Theorem

The linear code C is cyclic **iff** *C is an ideal in $\mathbb{F}[x]/(x^n-1)$.*

Constructing ideals - example

Example

Let $g(x) = x+1$ and we want to construct an ideal $\pmod{x^3+1}$.

$1\cdot(x+1)$	$x+1$	(110)
$x\cdot(x+1)$	$x+x^2$	(011)
$(1+x)(1+x)$	$1+x^2$	(101)
$x^2(1+x)$	$x^2+x^3=1+x^2$	(101)
$(1+x^2)(1+x)$	$1+x+x^2+x^3=x+x^2$	(011)
$(1+x+x^2)(1+x)$	$1+x^3=0$	(000)

We have a cyclic code (ideal) $C=\{(000),(110),(011),(101)\}$

Summary so far

So far we have established:

Facts

- The ring $\mathbb{F}[x]/(x^n - 1)$ is a principal ideal ring
- Linear code if cyclic iff C is an ideal in $\mathbb{F}[x]/(x^n - 1)$
- **Remains**: Find generators of ideals in $\mathbb{F}[x]/(x^n - 1)$

Generator polynomial

Theorem

Let $I \subset R$ and $g(x)$ a monic polynomial of least degree in I.
Then $g(x)$ generates I and $g(x)|f(x) = x^n - 1$

Proof.

(Sketch) $g(x)$ "obviously" generates I. As for the divisibility,

$$\begin{aligned} f(x) &= h(x)g(x) + r(x) \quad \deg(r) < \deg(g) \\ [f(x)] &= [h(x)][g(x)] + [r(x)] \end{aligned}$$

As $[f(x)] = 0$ then $[r(x)] \in I$, and $r(x) = 0$, i.e. $g|f$. □

Uniqueness of generator polynomial

Facts

There is a unique monic polynomial, $g(x)$ of least degree that generates I.

Justification

Indeed, if $g, h \in I$ and $deg(h) = deg(g)$ then

$$h(x) = a(x)g(x)$$

and $a(x) = 1$.

$g(x)$ is called generator polynomial

Uniqueness of generator polynomial II

KNOWLEDGE : Generator polynomial of some I must divide $f(x) = x^n - 1$

There is 1-1 correspondence between cyclic subspaces of $V_n(\mathbb{F})$

and monic polynomials $g(x) \in \mathbb{F}[x]$ that divide $f(x) = x^n - 1$.

Factorization - Example

Example

Consider $V_7(Z_2)$ and $f(x) = x^7 - 1$. Factorization of f over Z_2 is,

$$x^7 - 1 = (x+1)(x^3+x^2+1)(x^3+x+1).$$

Thus, the monic divisors of f are

$$\begin{aligned} g_1(x) &= 1 \\ g_2(x) &= x+1 \\ g_3(x) &= x^3+x^2+1 \\ g_4(x) &= (x+1)(x^3+x^2+1) \\ &\vdots \\ g_8(x) &= (x+1)(x^3+x^2+1)(x^3+x+1) \end{aligned}$$

Exactly 8 cyclic subspaces (follows from factorization).

Example cont.

Example

Take now

$$g_7(x) = (x^3+x^2+1)(x^3+x+1)$$

What is the dimension of S ?

$$S = \{(0000000), (1111111)\}.$$

Check:
$x(x^3+x^2+1)(x^3+x+1) = 1+x+x^2+\cdots+x^6 \bmod x^7-1$,
i.e. (1111111).

Repeat:$x(1+x+x^2+\cdots+x^6) \bmod x^7-1 = 1+x+x^2+\cdots+x^6$.

For $n = 7$ and $\deg(g) = 6$ we obtain $\dim(S) = 1 = 7-6$;
previously $\deg(g) = 1$ and $\dim(S) = 2$ for $n = 3$. Coincidence ?

Trivial subspaces

What if we take $g = 1$ as generator polynomial ?

We get $k = n$, that is the rate $R = 1$ no correcting/detecting capability !

What if we take $g = x^n - 1$ as a generator polynomial ?

We get $k = 0$ rate is zero, transmitting all zeros !

What if $g(x) = (x + 1)a(x)$ where $a(x)|x^n - 1$?

Come to exercises !

Divisors of $x^n - 1$

Remember that only the products in factorization of $x^n - 1$ can divide $x^n - 1$.

Codewords of the form $x^k g(x)$ does not divide $x^n - 1$.

Important to get confidence in these polynomials through exercises.

Example

Previously we used $g(x) = 1 + x$ to generate a cyclic subspace of $V_3(F_2)$. The factorization is,

$$x^3 + 1 = (1 + x)(1 + x + x^2)$$

Factorization of $x^n - 1$

Only consider here factorization over $GF(2)$. Do we have nontrivial cyclic subspaces for any n ?

There is one unfortunate choice $n = 2^r$ because $1 + x^{2^r} = (1 + x)^{2^r}$ so 1 is the only zero of $f(x)$ with multiplicity 2^r.

–Otherwise, use MAPLE, MAGMA ... and factor, works fine for practical sizes.

- REMARK: Sometimes only few possibilities, e.g. $1 + x^{24} = (1 + x)^8(1 + x + x^2)^8$

– Remains: finding cyclic subspace of a given dimension k !

Cyclic subspaces of dimension k

Theorem
Let g be a monic divisor of $x^n - 1$ over $\mathbb{F}$ with $\deg(g) = n - k$.

Then g is a generator polynomial for a cyclic subspace of $V_n(F)$ of dimension k

Cyclic subspaces of dimension k - proof

Proof.
Let S be the cyclic subspace of $V_n(\mathbb{F})$ generated by g.

CLAIM: $B = \{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ is a basis for S.

B linearly independent: Consider

$$\sum_{i=0}^{k-1} \lambda_i x^i g(x) = 0, \ \lambda_i \in \mathbb{F}$$

As $\deg(g) = n - k$ the only one term which contain x^{n-1} is λ_{k-1}, i.e. $\lambda_{k-1} = 0$. Same for λ_i, $0 \leqslant i \leqslant k-1$. □

Cyclic subspaces of dimension k - proof II

Proof.
B spans S: Let $h(x) \in S$, then $h(x) = a(x)g(x)$.

- W.l.o.g. we may assume that $\deg(a) < k$, otherwise mod reduction. Thus,

$$a(x) = \sum_{i=0}^{k-1} \lambda_i x^i \Rightarrow h(x) = \sum_{i=0}^{k-1} \lambda_i x^i g(x)$$

- So B spans S and $\dim(S) = k$.

□

Basis - example

Example

Want to construct a binary cyclic $(15, 9)$ code. Simply take

$$g(x) = (1 + x + x^2)(1 + x + x^4)$$

$deg(g) = 6$ and construct the basis

$$B = \{g(x), xg(x), \ldots, x^8 g(x)\}$$

Generator matrix of cyclic codes

Current knowledge: A generator polynomial $g(x)|x^n - 1$ of degree $n - k$ generates an (n, k) code C.

- Encoding: Codewords are products $a(x)g(x)$ where $\deg(a) \leqslant k - 1$.
- Message space is a set of polynomials in $\mathbb{F}[x]$ with degree $< k$.

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \\ \vdots \\ x^{k-1} g(x) \end{bmatrix}.$$

Generator matrix - example

Example

$g(x) = x^4 + x^2 + x + 1$ generates a cyclic (equidistant) (7,3) code. The generator matrix is,

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Encoding the message $a = (101)$ means adding the first and the third row of G , i.e. $aG = (1101001)$.

Check,

$$a(x)g(x) = (1 + x^2)g(x) = 1 + x + x^3 + x^6.$$

Cyclic codes - repetition

- How can cyclic subspaces be constructed ?

  Using factorization of $x^n - 1$

- For a given k can a k-dimensional subspace be constructed - if factorization

  $$x^n - 1 = p_1(x)^{a_1} p_2(x)^{a_2} \cdots p_t(x)^{a_t}$$

  allows finding $g|f$ such that $\deg(g) = n - k$;

- Then $\dim(C) = k$.

Dual code

A cyclic code C - fully specified with the generator matrix,

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}.$$

What about the dual code of C ? Is it cyclic ?

As you might guess the interesting polynomial is

$$h(x) = f(x)/g(x),$$

where G generates C and $f(x) = x^n - 1$.

Properties of $h = f/g$

Assume $\deg(g) = n - k$ and g is a monic divisor of $f(x) = x^n - 1$.

Facts

Then for $h = f/g$ we have:

- $\deg(h) = k$
- h is monic
- h generates a cyclic code C' of dimension $n - k$.

Is $h = f/g$ a dual code (parity check)

We continue our investigation:

Taking $c_1(x) = a_1(x)g(x) \in C$ and $c_2(x) = a_2(x)h(x) \in C'$ we have,

$$c_1(x)c_2(x) = a_1(x)a_2(x)f(x) \equiv 0 \pmod{f(x)}.$$

- So far everything fits
- Thus, C' is a dual of C ?

Dual code and zero divisors

The fact that $c_1 c_2 \equiv 0 \in \mathbb{F}[x]/f(x)$ does not imply $\mathbf{c}_1 \cdot \mathbf{c}_2 = 0$.

Example

Let $f(x) = x^7 - 1$. Over GF(2) it factors as,

$$x^7 - 1 = (1 + x)(1 + x + x^3)(1 + x^2 + x^3).$$

Then taking $c_1 = (1 + x)(1 + x + x^3) = 1 + x^2 + x^3 + x^4$ and $c_2 = 1 + x^2 + x^3$ we have,

$$\mathbf{c}_1 \cdot \mathbf{c}_2 = 1 + 1 + 1 = 1 \pmod{2}.$$

Dual of a cyclic code

Is there a connection between C' and $C^{\perp}$?

YES, they are equivalent codes !

Facts

- To get orthogonality of the codewords of C and C' it is enough to **reverse the order** of vectors in C' (see the textbook) !!

Coordinate reversing

- E.g. $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in C' \rightarrow \tilde{\mathbf{c}} = (c_n, c_{n-1}, \ldots, c_2, c_1)$
- Note that for fixed $g(x)$ the polynomial $h(x) = (x^n - 1)/g(x)$ is also fixed.
- $h(x)$ generates C' and the generator matrix of C' is,

$$G' = \begin{bmatrix} h(x) \\ xh(x) \\ x^2 h(x) \\ \vdots \\ x^{n-k-1} h(x) \end{bmatrix}.$$

Example - coordinate reversing

Example

In the previous example we had:

$$c_1 = (1+x)(1+x+x^3) = 1+x^2+x^3+x^4 \in C$$

and

$$c_2 = 1+x^2+x^3 \in C'$$

so that,

$$\mathbf{c}_1 \cdot \mathbf{c}_2 = 1+1+1 = 1 \pmod 2.$$

- We check the inner product for reversed $\tilde{\mathbf{c}_2} = (0001101)$,

$$\mathbf{c}_1 \cdot \mathbf{c}_2 = (1011100) \cdot (0001101) = 0 \pmod 2.$$

What about the generator matrix of $C^\perp$?

Generator matrix of the dual code

- Reversing coordinates in C' = reversing the columns of G'.

Example

Let $g(x) = 1+x^2+x^3+x^4$ and $h(x) = 1+x^2+x^3$. Then,

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ x^2g(x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

$$G' = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}; G^\perp = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Parity check matrix

So what is

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

Of course, it is parity check matrix of C, i.e. $G^{\perp} = H$; check that

$$G G^{\perp T} = 0$$

Reciprocal polynomial

- Need a compact description of the dual code : the operation of reversing corresponds to reciprocal polynomial of h,

$$h_R(x) = \sum_{i=0}^{k} a_{k-i} x_i, \quad \text{for } h(x) = \sum_{i=0}^{k} a_i x_i.$$

- We only use that $c_i \leftarrow c_{n-i}$
- Remark: $h_R(x) = x^k h(1/x)$ (easy exercise).

Reciprocal polynomial -example

Example

In the previous example C' was spanned by

$$\{h(x), xh(x), \ldots, x^{n-k-1}h(x)\}$$

where $h(x) = 1 + x^2 + x^3$.

Therefore, $C^{\perp}$ is spanned by,

$$\{x^3h(1/x), x^4h(1/x), \ldots, x^6h(1/x)\}$$

For instance,

$$x^3h(1/x) = x^3(1 + x^{-2} + x^{-3}) = 1 + x + x^3$$

the last row of $G^{\perp}$.

Encoding cyclic codes - framework

- To extract information bits efficiently it is desirable to have generator matrix in systematic form.
- Easier (faster) encoding and decoding.
- Quite simple derivation of $G = [R\ I_K]$ in the systematic form
- A bit complicated approach to encoding in the textbook:
  - No need to construct or store G
  - The details are optional (for interested students).

Generator matrix in systematic form

To construct G in systematic form:

Algorithm

1. Divide x^{n-k+i} by $g(x)$ for $0 \leqslant i \leqslant k-1$:
2. Division gives $x^{n-k+i} = q_i(x)g(x) + r_i(x)$ where $\deg(r_i) < \deg(g) = n-k$.
3. Therefore,
$$x^{n-k+i} - r_i(x) = q_i(x)g(x) \in C.$$
4. Take the coefficients corresponding to vectors $x^{n-k+i} - r_i(x)$ as the rows of a matrix, i.e. form $G = [R\ I_k]$.

Generator matrix in systematic form - example

Example

Let $g(x) = 1 + x + x^3$ and $f(x) = x^7 - 1$. Then compute,

$$\begin{aligned} x^3 &= (1)(x^3 + x + 1) + 1 + x \\ x^4 &= (x)(x^3 + x + 1) + x + x^2 \\ x^5 &= (x^2 + 1)(x^3 + x + 1) + 1 + x + x^2 \\ x^6 &= (x^3 + x + 1)(x^3 + x + 1) + 1 + x^2 \end{aligned}$$

So the generator matrix is,

$$G = [R\ I_4] = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Encoding a message $\mathbf{m} = (1011)$ gives $\mathbf{c} = \mathbf{m}G = (1100\ 1011)$

Encoding algorithm (optional see the textbook)

Let $\mathbf{g} = (g_0 g_1 \cdots g_{n-k-1})$ and let $(a_0 a_1 \cdots a_{k-1})$ denote the message symbols.
$\mathbf{s} = (s_0 s_1 \cdots s_{n-k-1})$ - parity check symbols to be found.

1. Set $s_j = 0, 0 \leqslant j \leqslant n-k-1$. Set $i = 1$.
2. If $a_{k-i} = s_{n-k-1}$ then
   for j from $n-k-1$ to 1, set $s_j = s_{j-1}$
   $s_0 = 0$.
   Otherwise
   for j from $n-k-1$ to 1, set $s_j = s_{j-1} + g_j$
   $s_0 = g_0$.
3. $i = i + 1$
4. If $i > k$, stop. Otherwise, go to 2.

Decoding - preliminaries

- Need a syndrome vector as for other linear codes
- Efficient decoding procedure due to nice algebraic structure of syndromes
- Furthermore, one can decode certain error patterns efficiently - error trapping
- Burst error correcting - used in many applications e.g. storage mediums, burst channels etc.

Parity check matrix

– Assume C is a cyclic code in systematic form

$$G = [R\ I_k]$$

where the rows of R are derived as,

$$r_i(x) = x^{n-k+i} - q_i(x)g(x),\ 0 \leqslant i \leqslant k-1.$$

i.e. r_i is a remainder when x^{n-k+i} is divided by $g(x)$.

A parity-check matrix for C is $H = [I_{n-k}\ -R^T]$, that is $GH = 0$.

Syndromes for cyclic codes

- Need a syndrome vector as for other linear codes
- Define the syndrome **s** in a standard way $H\mathbf{r}^T = \mathbf{s}$.

Polynomial interpretation (Th. 5.11):

The syndrome $s(x)$ is the remainder polynomial when $r(x)$ is divided by $g(x)$,

$$s(x) \equiv r(x) \pmod{g(x)}.$$

Recall that we are using $H = [I_{n-k}\ -R^T]$.

Syndrome computing - example

Example

Let $g(x) = 1 + x + x^3$ generates a binary cyclic (7,4) code. Then using $G = [R\ I_4]$, and $H = [I_3\ -R^T]$

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

Let $\mathbf{r} = (1011011)$. The syndrome of $\mathbf{r}$ is $Hr^T = (001)^T = \mathbf{s}$.

Divide $r(x) = 1 + x^2 + x^3 + x^5 + x^6$ by $g(x)$ to get,

$$r(x) = (x^3 + x^2 + x + 1)g(x) + x^2 \Rightarrow \mathbf{s} = x^2.$$

Syndrome - cyclic shift

Cyclic shifts of the syndromes - useful for fast decoding

Facts

From $s(x) \equiv r(x) \pmod{g(x)}$ we can deduce,

- For a cyclic shift of $r(x)$ that is for $xr(x)$ we have,

$$xr(x) = xq(x)g(x) + xs(x) \Rightarrow xr(x) \equiv xs(x) \pmod{g(x)}.$$

Syndrome - cyclic shift

The syndrome of $xr(x)$ (binary case) is treated in the textbook,

- Syndrome of $xr(x)$ is $xs(x)$ if $\deg(s) < n-k-1$
- Syndrome of $xr(x)$ is $xs(x) - s_{n-k-1}g(x)$ if $\deg(s) = n-k-1$

– Easy to generalize to the i-th cyclic shift $x^i r(x)$,

$$x^i r(x) \equiv x^i s(x) \pmod{g(x)}.$$

Syndrome - example

Example

For $g(x) = 1 + x + x^3$ we have,

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Let received vector $\mathbf{r} = (1011011)$. The syndrome is $H\mathbf{r}^T = (001)^T = \mathbf{s}$.

Compute the syndrome of a cyclic shift of $\mathbf{r}$ (i.e. $\mathbf{w}$=(1101101)),

- $H\mathbf{w}^T = (110)^T$ or
- using polynomial representation: $xs(x) - g(x) = 1 + x$.

Cyclic run

Definition

A cyclic run of length $k \leqslant n$ is a succession of k cyclically consecutive elements in an n-tuple.

Example

$\mathbf{e} = (00110100)$ has a cyclic run of four 0's.

Decoding - reminder

Reminder

A linear (n, k) code with $d = 2t + 1$ can correct up to t errors.

Suppose $H = [I_{n-k} \ -R^T]$, and $\mathbf{e}$ is an error pattern $wt(\mathbf{e}) \leqslant t$.

For $\mathbf{r} = \mathbf{c} + \mathbf{e}$,

$$\mathbf{s} = H\mathbf{r}^T = H(\mathbf{c} + \mathbf{e})^T = H\mathbf{e}^T.$$

Decoding ideas

If we let $\hat{\mathbf{e}} = (\mathbf{s}^T, \mathbf{0})$ where $\mathbf{0}$ is all-zero k-tuple, then

$$H\hat{\mathbf{e}}^T = \mathbf{s}.$$

Facts

1. We know that $\hat{\mathbf{e}}$ and $\mathbf{e}$ are in the same coset of C, Th. 3.10
2. Suppose that $wt(\mathbf{s}) \leqslant t$ then $wt(\hat{\mathbf{e}}) \leqslant t$
3. Hence $\mathbf{e} = \hat{\mathbf{e}}$ - unique element in each coset of C of weight $\leqslant t$

The error is known to be $\mathbf{e} = (\mathbf{s}^T, \mathbf{0})$!!

Error trapping - idea

What about the syndromes of cyclic codes ?

Idea

– Again assume $wt(\mathbf{e}) \leqslant t$ having a cyclic run of at least k zeros.

Then the following can be deduced:

- We can find $0 \leqslant i \leqslant n-1$, so that the shift of $\mathbf{e}$ by i positions have all nonzero components in the first $n-k$ positions.

Example

$\mathbf{e} = (0011100)$ has a cyclic run of four 0's. Then shift 5 positions,

$$\mathbf{e}^{sh} = (1110000)$$

Error trapping - idea II

Facts

- **For this** i, $wt(s_i(x)) \leqslant t$ where

$$s_i(x) = x^i e(x) \pmod{x^n - 1}.$$

- Compute $s_i(x) = x^i r(x) \pmod{g(x)}$
- Then when $wt(s_i(x)) \leqslant t$ we have,

$$x^i e(x) = (\mathbf{s}_i, \mathbf{0}) \Rightarrow e(x) = x^{n-i}(\mathbf{s}_i, \mathbf{0}),$$

where

$$x^{n-i}(\mathbf{s}_i, \mathbf{0}) = x^{n-i} s_i(x) \pmod{x^n - 1}$$

In words, $x^{n-i}(\mathbf{s}_i, \mathbf{0})$ is a shift of $(\mathbf{s}_i, \mathbf{0})$ $n - i$ positions.

Decoding algorithm - example

Example

$g(x) = 1 + x^2 + x^3$ generates $(7, 4)$-cyclic code with $d = 3$.

Consider $c(x) = a(x)g(x)$ for $a(x) = 1 + x + x^2$ so that,

$$c(x) = 1 + x + x^5.$$

Suppose $e(x) = x^6$ is introduced hence $r(x) = 1 + x + x^5 + x^6$.

First compute the syndrome of $r(x)$,

$$r(x) = (x^3 + 1)g(x) + \overbrace{(x + x^2)}^{s_0(x)}.$$

Decoding algorithm - example cont.

Example (Cont.)

Since $wt(s_0) = 2$ we compute the syndrome of $xr(x)$ (using $s_0(x)$),

$$s_1(x) = xr(x) \pmod{g(x)} = x(x + x^2) = 1 \pmod{1 + x^2 + x^3}.$$

Since $wt(s_1) \leqslant 1 = t$ we find the error pattern,

$$e(x) = x^{n-1}s_1(x) \pmod{x^n - 1} = x^6.$$

ERROR CORRECTION: Decode using

$$c(x) = r(x) + x^6 = 1 + x + x^5$$

Decoding algorithm (Error trapping)

Input

– t-error correcting (n, k) cyclic code with $g(x)$;

– $e(x)$ error pattern, $wt(e) \leqslant t$, **cyclic run of at least k 0's.**

1. Compute the syndrome $s_0(x)$ of $r(x)$ using EA,
$$r(x) = q(x)g(x) + s_0(x). \quad \text{Set } i = 0.$$
2. If $wt(s_i) \leqslant t$, set $e(x) = x^{n-i}(\mathbf{s}_i, \mathbf{0})$, decode to $r(x) - e(x)$.
3. Set $i = i + 1$.
4. If $i = n$ then stop; the error pattern is not trappable.
5. If $\deg(s_{i-1} \leqslant n - k - 1$ then set $s_i(x) = xs_{i-1}(x)$
   otherwise set $s_i(x) = xs_{i-1}(x) - g(x)$
6. Go to 2.

Decoding algorithm - trapping error patterns

- In the previous example any single error implies the existence of a cyclic run of six 0's.

  Since $6 \geqslant k = 4$ the algorithm never fails !

- Is this always the case ?
- NOT in general, but sometimes simple combinatorics ensures the success of the algorithm.

Trapping error patterns I

Example

- For instance, for a cyclic (15,7) binary code suppose $d = 5$.
- Then any error pattern of weight at most 2 in the vector of length 15 must contain a run of length at least 7 !

$$(0000001000010000) \rightarrow (1000010000000000)$$

- In other words, all single and double error can be corrected.

Trapping error patterns II

Example

- Consider a cyclic (15,5) binary code generated by

$$g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$$

and suppose $d = 7$.

- Then any error pattern of weight at most 3 in the vector of length 15 must contain a run of length at least 5 zeros unless

$$\hat{\mathbf{e}} = (100001000010000)$$

- We can correct all errors of weight $\leqslant 3$ but not the pattern above and its cyclic shifts !

Read Example 18 (pg. 174) in details.

Burst error correcting

– So far we have considered random error patterns.

– If the errors come in bursts - a special approach to correct such errors.

Definition

: A cyclic burst error of length t - vector with nonzero components within a cyclic run of length t; the first and the last component in the run being nonzero.

Example

- $\mathbf{e}_1 = (0101\ 0110\ 000)$ is a burst of length 6.
- $\mathbf{e}_2 = (0000\ 0010\ 001)$ is a burst of length 5.

Polynomial description

Polynomial description:

$$e(x) = x^i b(x) \pmod{x^n - 1},$$

where $b(x)$ describes the error pattern and i indicates where the burst begins.

Example

$$\mathbf{e}_1 = (0101\ 0110\ 000) \leftrightarrow e(x) = x(1 + x^2 + x^4 + x^5)$$

Burst error correcting - Theorem

Theorem

A linear code C can correct all burst errors of length t or less **iff** all such errors occur in distinct cosets of C.

Justification

- If all burst errors are in distinct cosets of a standard array for C then unique identification via syndromes - errors correctable .
- Suppose $\mathbf{b}_1$ and $\mathbf{b}_2$ in a coset C_i of C. Then

$$\mathbf{b}_1 - \mathbf{b}_2 = \mathbf{c} \neq \mathbf{0}.$$

- Cannot decode $\mathbf{b}_1$ if it is a received vector ? Either $\mathbf{0} \rightarrow \mathbf{b}_1$; decode as $\mathbf{0}$.
- OR : But also $\mathbf{c} \rightarrow \mathbf{b}_1$, by error $\mathbf{b}_2$; decode as $\mathbf{c}$.

Burst error detection

- Determining the length of the correctable burst is a combinatorial challenge.
- Read Example 20 and discussion there.
- In this example a cyclic (15,9) code with $d = 5$ can correct all bursts of length $\leqslant 3$.
- "Normal" correcting capability is 2 errors (random position).
- **Main idea:** Syndrome polynomials of all possible burst patterns of length $\leqslant t$ are different.

Error trapping for burst error codes

Can be shown that a t-burst error correcting (n, k) code satisfies

$$n - k \geqslant 2t$$

Therefore, $n - t \geqslant k$.

Moral : A burst error of length t has a cyclic run of $n - t$ 0's, i.e. at least cyclic run of k 0's - requirement for error-trapping algorithm.

Can use the error trapping algorithm.

Error trapping for burst error codes II

Algorithm

– t-error **burst** correcting cyclic code with $g(x)$;

For t-error correcting code we had:

2. If $wt(s_i) \leqslant t$, set $e(x) = x^{n-i}(\mathbf{s}_i, \mathbf{0})$, decode to $r(x) - e(x)$.

For t-error burst correcting code we have:

2. If s_i is a burst of length t or less, then $e(x) = x^{n-i}(\mathbf{s}_i, \mathbf{0})$, decode to $r(x) - e(x)$.

Chapter 4

Cyclic codes with designed minimum distance

Contents of the chapter:

- Factoring in finite fields Minimal polynomials
- Factoring $x^n 1$
- Roots
- BCH codes
- Bounds for cyclic codes

Preliminaries

- Cyclic subspaces are constructed using factorization of $x^n - 1$
- Factoring in finite fields - a good preparation for BCH codes
- Need somewhat more advanced results from finite fields
- Eventually, we get confident with:
  - Primitive roots of unity
  - Cyclotomic cosets, minimal polynomials etc.

Table of contents

Minimal polynomials - preliminaries

Definition

A minimal polynomial of $\beta \in \mathbb{F}_{p^k}$ with respect to $\mathbb{F}_p$ is a monic polynomial $m(x)$ of least degree such that $m(\beta) = 0$

Example

Recall our first field $\mathbb{F}_{2^2} = \{0, 1, \alpha, \alpha^2 = \alpha + 1\}$ where α is a primitive root of polynomial $x^2 + x + 1$.

$$\begin{aligned} 1 \quad & m(x) = x + 1 \\ \alpha \quad & m(x) = x^2 + x + 1; \ \ \alpha^2 + \alpha + 1 = 0 \\ \alpha + 1 \quad & m(x) = x^2 + x + 1; \ \ (\alpha + 1)^2 + \alpha + 1 + 1 = 0 \end{aligned}$$

How do we find minimal polynomials ?

Minimal polynomials - properties

Facts

- The minimal polynomial of an element α is unique (see the textbook)
- The minimal polynomial m_α of an element $\alpha \in \mathbb{F}_q^*$ is irreducible.

Definition

For $\alpha \in \mathbb{F}_q$ let t be the smallest positive integer s.t. $\alpha^{p^t} = \alpha$. The set,

$$C(\alpha) = \{\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{t-1}}\}$$

is called the conjugates of α w.r.t. $\mathbb{F}_p$. Note $C(\alpha) = C(\alpha^{p^i})$.

Finite field $Z_2[x]/x^3 + x + 1$- reminder

- Construction of $\mathbb{F}_{2^3}$ using $f(x) = x^3 + x + 1$.

$$\begin{aligned}
x^0 &= 1 && (100)\\
x^1 &= x && (010)\\
x^2 &= x^2 && (001)\\
x^3 &= x+1 \pmod{x^3+x+1} && (110)\\
x^4 &= x^2+x \pmod{x^3+x+1} && (011)\\
x^5 &= x^2+x+1 \pmod{x^3+x+1} && (111)\\
x^6 &= x^2+1 \pmod{x^3+x+1} && (101)\\
x^7 &= 1 \pmod{x^3+x+1} && (100)
\end{aligned}$$

Denote this element $x = \alpha$ - any element in $\mathbb{F}_{2^3}$ is power of α; and $\alpha^7 = 1$.

Back to factoring

What are the roots of $x^7 - 1$?

From the previous example (and Lagrange theorem) any element β in $\mathbb{F}_{2^3}$ satisfies $\beta^7 = 1$!!

This means that (all) the roots of $x^7 - 1$ are the (nonzero) elements of $\mathbb{F}_{2^3}$.

Thus we can write,

$$x^7 - 1 = \prod_{\beta \in \mathbb{F}_{2^3}^*} (x - \beta).$$

Conjugates - example

Example

Consider the element $\alpha \in \mathbb{F}_{2^3}$. Its conjugates over GF(2) are :

$$C(\alpha) = \{\alpha, \alpha^2, \alpha^4, \alpha^8 = \alpha\}$$

If we take α^3 then

$$C(\alpha^3) = \{\alpha^3, \alpha^6, \alpha^{12} = \alpha^5, \alpha^{10} = \alpha^3\}$$

Example

Assume α is a root of $g(x) = x^3 + x + 1$. Then,

$$g(\alpha^2) = (\alpha^2)^3 + \alpha^2 + 1 = (\alpha^3 + \alpha + 1)^2 = 0.$$

Minimal polynomials - roots

Lemma (2.10)

Let $\alpha \in \mathbb{F}^*$ and $\text{char}(\mathbb{F}) = p$. Then,

$$m_\alpha(x) = \prod_{\beta \in C(\alpha)} (x - \beta)$$

is a minimal polynomial of α. The coefficients are in $GF(p)$!!

Minimal polynomials - example

Example

Consider the field $\mathbb{F}_{2^3} = \mathbb{F}_2[x]/(x^3 + x + 1)$. Let us compute min. polynomial of α^3 ($= x + 1$ as a polynomial). Remark $\alpha^7 = 1$.

Cyclotomic coset $C(\alpha^3) = \{\alpha^3, \alpha^6, \alpha^{12} = \alpha^5\}$

$$\begin{aligned} m_{\alpha^3}(x) &= \prod_{\beta \in C(\alpha^3)} (x - \beta) = \prod_{\beta \in \{\alpha^3, \alpha^5, \alpha^6\}} (x - \beta) \\ &= (x - \alpha^3)(x - \alpha^5)(x - \alpha^6) = x^3 + (\alpha^{15} + \alpha^{18} + \alpha^{30})x^2 \\ &+ (\alpha^3 + \alpha^5 + \alpha^6)x + \alpha^{14} = x^3 + x + 1 \end{aligned}$$

Check that $\alpha^{15} + \alpha^{18} + \alpha^{30} = \alpha + \alpha^4 + \alpha^2 = 0$.

Factoring $x^n - 1$ over extension fields

Goal

: Factor $x^n - 1$ over $GF(q)$ with the prime field of characteristic p.

Example

Cannot factor $x^2 + 1$ over $\mathbb{R}$. But it factors nicely over $\mathbb{C}$ as

$$(x - i)(x + i)$$

Definition

Define m to be the order of q, i.e. the least integer s.t.

$$q^m \equiv 1 \pmod{n} \Leftrightarrow q^m - 1 = kn.$$

Factoring $x^n - 1$ -example

Example

Want to factor $x^5 - 1$ over GF(2), i.e. $n = 5$ and $q = 2$.

Need to find m satisfying $2^m \equiv 1 \pmod 5$!

Smallest $m = 4$; in other words $x^5 - 1$ factors in $GF(2^4)$!

If α is a primitive element (generator) of $GF(2^4)$ we know $\alpha^{15} = 1$.

The roots of $x^5 - 1$ are:

$$1, \alpha^3, \alpha^6, \alpha^9, \alpha^{12}.$$

Factoring $x^n - 1$ -example

Example (Cont.)

Therefore,

$$x^5 - 1 = (x - 1)(x - \alpha^3)(x - \alpha^6)(x - \alpha^9)(x - \alpha^{12}).$$

Not factoring over GF(2) ?

It turns out that the coefficients of

$$(x - \alpha^3)(x - \alpha^6)(x - \alpha^9)(x - \alpha^{12})$$

belongs to $GF(2)$!!

Splitting field

- Using the fact (geometric series)

$$x^n + x^{2n} + \cdots + x^{kn} = \frac{x^n(x^{kn} - 1)}{x^n - 1},$$

to know that $x^n - 1 | x^{kn} - 1 = x^{q^m - 1} - 1$.

- Denote $\mathbb{F} = GF(q^m)$; we know that any $\alpha \in \mathbb{F}$ is a root of $x^{q^m} - x$ (since $\alpha^{q^m} = \alpha$).
- Thus, all the roots of $x^n - 1$ are in $\mathbb{F}$, which is called a **splitting field** of $x^n - 1$ over $GF(q)$.

Primitive roots of unity

Definition

If $\gamma \in \mathbb{F}$ is a root of $x^n - 1$, i.e. $\gamma^n = 1$, then γ is called an n^{th} root of unity.

Facts

- Suppose α is a primitive element of $\mathbb{F}$, then α has (multiplicative) order $q^m - 1$.
- Then $\alpha^k = \alpha^{(q^m-1)/n}$ has order n and it is root of $x^n - 1$.
- α^k is called a primitive n^{th} root of unity as $(\alpha^k)^n = 1$ and $(\alpha^k)^n \neq 1$ for all $j < n$.

Cyclotomic cosets

Definition

Given q and n and a fixed integer $0 \leqslant i \leqslant n-1$ a cyclotomic coset containing i is defined as,

$$C_i = \{i, iq, iq^2, \ldots, iq^{s-1}\} \bmod n,$$

where s is the smallest integer s.t. $iq^s \equiv i \pmod{n}$.

Facts

- An equivalence relation on integers $[0, n-1]$. The set of cyclotomic cosets of q modulo n is

$$\mathrm{C} = \{C_i : 0 \leqslant i \leqslant n-1\}$$

.

Cyclotomic cosets -example

Example

Let $n = 9$, $q = 2$. Then,

$$\begin{aligned} C_1 &= \{1,2,4,8,7,5\} = C_2 = C_4 = C_8 = C_5 = C_7 \\ C_3 &= \{3,6\} = C_6 \\ C_0 &= \{0\} \end{aligned}$$

Minimal polynomials over $GF(q)$

Equivalent to the definition of min. polynomials over $GF(p)$.

Facts

Minimal polynomial of $\beta \in GF(q^m)$ over $GF(q)$ satisfies:

- **Monic** polynomial $m(x) \in GF(q)[x]$ of **least degree** s.t. $m(\beta) = 0$
- **Unique and irreducible**
- Conjugates of β are :$\beta^q, \beta^{q^2}, \ldots, \beta^{q^{t-1}}$ with t least integer $\beta^{q^t} = \beta$
- The minimal polynomial of β is, $m_\beta(x) = \prod_{i=0}^{t-1}(x - \beta^{q^i})$.

Roots of $x^n - 1$

Assume: α is a primitive element of $GF(q^m)$ and $q^m - 1 = kn$ so that α^k is a primitive n^{th} root of unity; $\alpha^n = 1$.

Facts

- $x^n - 1$ has n distinct zeros $(\alpha^k)^i$ for $i = 0, 1, \ldots, n-1$ as,

$$(\alpha^{ki})^n = (\alpha^{kn})^i = 1^i = 1.$$

- Now if $\beta = \alpha^{ki}$ is a root of $x^n - 1$ so are the conjugates of β,

$$(\beta^{q^j})^n = (\beta^n)^{q^j} = 1$$

Factoring $x^n - 1$ II

Facts

- Therefore, $m_\beta(x) | x^n - 1$ and the roots of $m_\beta(x)$ are

$$\alpha^{ki}, \alpha^{kiq}, \ldots, \alpha^{kiq^{t-1}},$$

where t is the smallest integer s.t.

$$kiq^t \equiv ki \pmod{kn} \quad iq^t \equiv i \pmod{n}$$

Thus, the degree of $m_\beta(x)$ is the cardinality of C_i.

The number of irreducible factors of $x^n - 1$ is the number of cosets in the partition of $[0, n-1]$

Factoring $x^n - 1$- example

Example

Suppose we wish to factor $f(x) = x^{15} - 1$ over $GF(2)$.

Then $m = 4$ as $2^4 \equiv 1 \pmod{15}$. The cyclotomic cosets are:

$$\begin{aligned} C_0 &= \{0\}; \quad C_1 = \{1, 2, 4, 8\} \\ C_3 &= \{3, 6, 9, 12\}; \quad C_5 = \{5, 10\} \\ C_7 &= \{7, 14, 13, 11\} \end{aligned}$$

Thus, $x^{15} - 1$ factors into one linear term, one irreducible quadratic and 3 irreducible quartics.

We have to find $m_\alpha(x), m_{\alpha^3}(x), m_{\alpha^5}(x), m_{\alpha^7}(x)$.

Factoring $x^n - 1$- example cont.

Example (Cont.)

Simply compute e.g.

$$m_\alpha(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8)$$

using $1 + x + x^4$ to generate the field $GF(2^4)$, i.e. $\alpha^4 = \alpha + 1$.

As expected (why?) $m_\alpha(x) = 1 + x + x^4$.

Continue and compute $m_{\alpha^3}(x), m_{\alpha^5}(x), m_{\alpha^7}(x)$ and trivially $m_1(x) = x - 1$.

$$x^{15}-1 = (x-1)(1+x+x^4)(1+x^3+x^4)(1+x+x^2)(1+x+x^2+x^3+x^4)$$

Splitting field of a polynomial

Definition

Splitting field of $g(x) \in GF(q)[x]$ is the smallest field $GF(q^m)$, for some m, that contain all the roots of $g(x)$.

In other words $g(x)$ splits in $GF(q^m)$ so that,

$$g(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_n);\ \alpha_i \in GF(q^m).$$

- Splitting field always exists (can be determined from the degrees of its irreducible factors over $GF(q)$).

Splitting field of a polynomial - example

Example

For example,

$$g(x) = x^2 + x + 1$$

has no roots in $GF(2)$ but it splits in $GF(2^2)$!!

Let $GF(2^2) = \{0, 1, \alpha, \alpha^2\}$, the field generated by $x^2 + x + 1$ so that $\alpha^2 + \alpha + 1 = 0$.

Then,

$$g(x) = (x + \alpha)(x + \alpha^2) = x^2 + x\underbrace{(\alpha + \alpha^2)}_{1} + \underbrace{\alpha^3}_{1}.$$

Means: the roots of g are α and α^2 !

BCH codes - motivation

Motivation

- So far nothing has been said about the minimum distance of cyclic codes.
- Recall that given a code (list of codewords) determining the minimum distance is an NP-hard problem.
- In general given **some** generator polynomial of cyclic code we do not know what is the minimum distance (n large).

BCH codes - motivation II

Goal

- Our goal is to relate the roots of a generator polynomial to minimum distance.
- We construct (select) a generator polynomial with a prescribed set of roots !

Finding the splitting field - example

Example

Consider the polynomial over $GF(2)$,

$$g(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}.$$

How do we find the splitting field ? At the moment by checking !

No roots in $GF(2)$, $GF(2^2)$, $GF(2^3)$ and $GF(2^4)$.

Let us try $GF(2^5)$ generated by

$$h(x) = 1 + x^2 + x^5,$$

i.e. α is a root of $1 + x^2 + x^5$.

Finding the splitting field - example cont

Example

- Then both α and α^3 are roots of $g(x)$
- Furthermore the conjugates of α and α^3,

$$\alpha^2, \alpha^4, \alpha^8, \alpha^{16} \quad \text{and } \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17},$$

  are also the roots of $g(x)$.

- Thus, $g(x)$ splits in $GF(2^5)$ - cannot have more than 10 roots !!
- Note that $\alpha, \alpha^2, \alpha^3$ are three consecutive roots of $g(x)$.

The splitting field of generator polynomial

- Need a connection to cyclic codes.
- Let $g(x)$ be a generator for a cyclic (n, k)-code C over $\mathbb{F} = GF(q)$.
- The degree of g is $n - k$ so g splits in some $GF(q^m)$ as,

$$g(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-k}); \ \alpha_i \in GF(q^m).$$

- If $m_{\alpha_i}(x)$ is a minimal polynomial of α_i then $m_i | g$ since

$$m_{\alpha_i}(x) = \prod_{\beta \in C(\alpha_i)} (x - \beta)$$

.

Roots of the codewords

Recall that a cyclic code C is an ideal generated by $g(x)$, i.e.

$$C = \{g(x)a(x) : a(x) \in R\}$$

Facts

- A polynomial $c(x)$ is in C iff $g(x)|c(x)$. Then if $c(x) \in C$,

$$c(\alpha_i) = 0 \text{ for } 1 \leqslant i \leqslant n-k$$

Roots of cyclic codes - main theorem

Theorem

Suppose $g(x)$ generates a cyclic (n, k)-code C over $\mathbb{F} = GF(q)$ and has roots

$$\alpha_1, \alpha_2, \ldots, \alpha_{n-k}$$

in some $GF(q^m)$.

Then $t(x) \in C$ iff $t(\alpha_i) = 0$ for all i, $1 \leqslant i \leqslant n-k$.

Our goal is to find parity check matrix of C using association,

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n \leftrightarrow c(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}$$

Orthogonal matrix

Let us consider the following matrix associated to C,

$$H = \begin{bmatrix} \alpha_1^0 & \alpha_1^1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ \alpha_2^0 & \alpha_2^1 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_{n-k}^0 & \alpha_{n-k}^1 & \alpha_{n-k}^2 & \cdots & \alpha_{n-k}^{n-1} \end{bmatrix}$$

This $(n-k) \times n$ matrix over $GF(q^m)$ is orthogonal to any codeword in C !

$$H_i \mathbf{c}^T = \sum_{j=0}^{n-1} \alpha_i^j c_j = c(\alpha_i) = 0.$$

Orthogonal matrix of cyclic code

Note that H is not a parity check matrix of C over $GF(q)$ though the following is valid,

$$c \in C \text{ iff } H\mathbf{c}^T = 0.$$

To get a parity check matrix over $GF(q)$ from H we need to represent elements of $GF(q^m)$ as m-tuples over $GF(q)$?

Parity check matrix of cyclic code

Algorithm

INPUT: H a matrix over $GF(q^m)$; each $\alpha_i \in GF(q^m)$

1. Replacing α_i^j with **column vectors** of length m over $GF(q)$
2. Obtain an $m(n-k) \times n$ matrix over $GF(q)$!
3. One can show that rows of H over $GF(q)$ are also orthogonal to the codewords of C **but** the rows are no longer independent over $GF(q)$.
4. Remove dependent rows to get a (standard) parity check matrix of C !

Parity check matrix - example

Example

Consider $g(x) = 1 + x + x^3$ that generates a binary $(7, 4)$ code. All the roots of $g(x)$ lie in $GF(2^3)$.

If α is a primitive element of $GF(2^3)$ generated by $x^3 + x + 1$, then α, α^2 and α^4 are roots of g.

But these roots are conjugates of each other. Therefore,

$$H = [1\ \alpha\ \alpha^2\ \alpha^3\ \alpha^4\ \alpha^5\ \alpha^6] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Parity check matrix - another example

Example

- Consider $g(x) = 1 + x + x^2 + x^3$ that generates a binary $(8, 5)$ code over $GF(3)$. Note that,

$$x^8 - 1 = (x^2 + 2x + 2)(x^2 + 1)(x + 1)(x + 2)(x^2 + x + 2).$$

- All the roots of $g(x)$ lie in $GF(3^2)$ (check that $0, 1, 2$ are not roots of g).
- Let $GF(3^2)$ be constructed using $f(x) = 2 + x + x^2$. Then the roots of g are: $\alpha^2, \alpha^4, \alpha^6$.

$$g(\alpha^2) = 1 + \alpha^2 + \alpha^4 + \alpha^6 = 1 + (1 + 2\alpha) + (2) + (2 + \alpha) = 0.$$

Parity check matrix - another example II

Example (cont.)

The parity matrix over $GF(3^2)$ is,

$$H = \begin{bmatrix} (\alpha^2)^0 & (\alpha^2)^1 & \cdots & (\alpha^2)^7 \\ (\alpha^4)^0 & (\alpha^4)^1 & \cdots & (\alpha^4)^7 \\ (\alpha^6)^0 & (\alpha^6)^1 & \cdots & (\alpha^6)^7 \end{bmatrix}$$

But the minimal polynomial of α^6 the same as for α^2 ! Therefore,

$$H = \begin{bmatrix} (\alpha^2)^0 & (\alpha^2)^1 & \cdots & (\alpha^2)^7 \\ (\alpha^4)^0 & (\alpha^4)^1 & \cdots & (\alpha^4)^7 \end{bmatrix}$$

Parity check matrix - another example III

Example (cont.)

But H is not a parity check for C. Need to go down to the field $GF(3)$.

Replace $(\alpha^i)^j$ as 2-tuples where each coordinate is in $GF(3)$, i.e. $\alpha = (0,1)$, $\alpha^2 = 1 + 2\alpha = (1,2)$ etc. The resulting matrix is,

$$H' = \begin{bmatrix} 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 0 & 2 & 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Remove the last row, then H' is a parity check of C. C has distance 2 ! (Why ?)

BCH codes - notation

Important class of cyclic codes was discovered in 1959 and 1960 independently by Bose & Chaudhuri and Hocquenghem - BCH codes.

– Notation :

- $lcm\{a(x), b(x), \ldots\}$ - the **least common multiple**
- $m_i(x)$ denotes the minimal polynomial of β^i, for $\beta^i \in GF(q^m)$.

BCH codes - definition

Definition

A **BCH code** over $\mathbb{F} = GF(q)$ of block length n and **designed distance** δ is a cyclic code generated by a polynomial

$$g(x) = lcm\{m_i(x) : a \leqslant i \leqslant a + \delta - 2\} \in \mathbb{F}[x]$$

whose root set contains $\delta - 1$ **distinct elements**

$$\beta^a, \beta^{a+1}, \ldots, \beta^{a+\delta-2}.$$

Here β is a primitive n^{th} root of unity and a is some integer.

BCH code - example

Example

Sometimes it turns out that the construction is "easy".

We consider again the Hamming (7,4)-code, that can be viewed as a cyclic code generated by

$$g(x) = 1 + x + x^3.$$

Among the roots of $g(x)$ are α and α^2, where α is a 7^{th} root of unity.

This means that our parameter $a = 1$ and this code is BCH-code with designed distance $\delta = 3$.

Special BCH codes

If $n = q^m - 1$ for some m the code is called primitive.

If $a = 1$ the code is called narrow-sense.

So our code in the previous example is both primitive and narrow-sense.

The BCH bound

One of the most important results for cyclic codes.

Theorem (6.2)

Let C be a BCH code over $GF(q)$ with designed distance δ.

Then C has minimum distance $\geqslant \delta$.

The BCH bound -proof

Proof.
We consider the matrix,

$$H = \begin{bmatrix} (\alpha^a)^0 & (\alpha^a)^1 & \cdots & (\alpha^a)^{n-1} \\ (\alpha^{a+1})^0 & (\alpha^{a+1})^1 & \cdots & (\alpha^{a+1})^{n-1} \\ (\alpha^{a+\delta-2})^0 & (\alpha^{a+\delta-2})^1 & \cdots & (\alpha^{a+\delta-2})^{n-1} \end{bmatrix}$$

formed by a **subset of roots** $\alpha^a, \alpha^{a+1}, \ldots, \alpha^{a+\delta-2}$.

The idea is to prove that no $\delta - 1$ columns of H are linearly dependent over $GF(q^m)$.

Then replacing $(\alpha^i)^j$ by the corresponding m-tuple, no $\delta - 1$ columns of are linearly dependent over $GF(q)$.

□

The BCH code - construction

In the previous example a (7,4)-code happened to be a BCH code.

Now we consider the design method when the code length is given.

Example

Designing problem: Suppose we wish to construct a BCH code of length 15 and designed distance 7 over $GF(2)$.

First we require a 15^{th} primitive root of unity α !

Solution: Can take a primitive element of $GF(2^4)$.

The BCH code - construction II

- Next we need minimal polynomials to define our $g(x)$.
- For $n = 15$ the cyclotomic cosets of $[0, n-1]$ are:

$$\begin{aligned} C_0 &= \{0\}; \quad C_1 = \{1, 2, 4, 8\} \\ C_3 &= \{3, 6, 9, 12\}; \quad C_5 = \{5, 10\} \\ C_7 &= \{7, 14, 13, 11\} \end{aligned}$$

- We know (from before) that $m_1(x), m_3(x), m_7(x)$ are of degree 4
- $m_5(x)$ is of degree 2 (check the length of $C_i's$!).

The BCH code - example cont.

Example (Cont.)

Taking

$$g(x) = m_1(x)m_3(x)m_5(x)$$

then $g|x^{15} - 1$ and $\deg(g) = 10$ so g generates a (15,5)-cyclic code.

The set of roots of $g(x)$ is,

$$R = \{\alpha^i : i \in \{1, 2, 4, 8, 3, 6, 9, 12, 5, 10\}\}.$$

This set contains **consecutive powers**

$$\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$$

so that C is a BCH code with designed distance 7.

Generalization of the bound

- BCH bound is valid only for BCH codes, not for general cyclic codes
- Instead of giving the emphasis on a code being cyclic we may explore the fact that cyclic codes are linear
- For linear codes the minimum distance of the code is the minimum weight of nonzero codewords
- Transformed into cyclic code representation - need to find a nonzero polynomial (codeword) of minimum weight (number of terms)
- BCH bound is then just a special case of this bound.

"Independent" family of subsets

Definition

Let S be a particular subset of $\mathbb{F} = GF(q^m)$; where all the roots of $x^n - 1$ are in $\mathbb{F}$.

Denote by I_S the **family of subsets of** $\mathbb{F}$ defined inductively by the rules:

1. $\emptyset \in I_S$
2. If $A \in I_S$, $A \subseteq S$ and $b \notin S$, then $A \cup \{b\} \in I_S$.
3. If $A \in I_S$ and $c \in \mathbb{F}$, $c \neq 0$, then $cA = \{ca : a \in A\} \in I_S$.

"Independent" family of subsets - example

Example

Consider $\mathbb{F} = GF(2^4)$ and let $S = \{\alpha, \alpha^2, \alpha^5, \alpha^7\}$

- $\emptyset \in I_S$
- $A_1 = \emptyset \cup \{\alpha^3\} = \{\alpha^3\} \in I_S$
- $A_2 = \alpha^{-1}A_1 \cup \{\alpha^3\} = \{\alpha^2, \alpha^3\} \in I_S$ since $A_1 \in I_S$ and $\alpha^{-1}A_1 \subseteq S$.
- $A_3 = \alpha^{-1}A_2 \cup \{\alpha^{11}\} = \{\alpha, \alpha^2, \alpha^{11}\} \in I_S$ since $A_2 \in I_S$ and $\alpha^{-1}A_2 \subseteq S$.

Main result - bound for cyclic codes

Each member of I_s is said to be *independent with respect to S*

Theorem

Let $h(x)$ be a polynomial over $GF(q)$ and let S be the set of roots of $h(x)$ in $GF(q^m)$ (not necessarily all roots of h are in $GF(q^m)$).

Then the number of nonzero terms in $h(x)$, denoted $wt(h(x))$ satisfies,

$$wt(h(x)) \geqslant |A|,$$

for every subset A of S which is independent with respect to S.

This bound improves the BCH bound. Read the proof for clarity (not exam question).

Computing the bound - example

Example

Consider binary cyclic code of length $n = 17$. The cosets are:

$$\begin{aligned} C_0 &= \{0\}; \\ C_1 &= \{1, 2, 4, 8, 16, 15, 13, 9\} \\ C_3 &= \{3, 6, 12, 7, 14, 11, 5, 10\} \end{aligned}$$

Suppose $g(x) = m_\alpha(x)$ where α is a primitive 17^{th} root of unity.

- According to BCH bound $g(x)$ has distance $d \geqslant 3$ (α, α^2 consecutive roots)
- We want to derive a better bound by showing that $wt(c) \geqslant 5$ for all $c \in C$, $c \neq 0$.

Computing the bound - example cont.

Example (Cont.)

Define the sequence $(A_i : i \geqslant 0)$ of sets independent w.r.t. $S = \{\alpha^i : i \in C_1\}$ by $A_0 = \emptyset$, $A_{i+1} = c_i A_i \cup \{b_i\}$. Through "smart" selection of b_i, c_i we have:

$$\begin{array}{ll} A_0 = \emptyset & \\ A_1 = \alpha^3 & c_1 = \alpha \quad b_1 = \alpha^{14} \\ A_2 = \{\alpha^4, \alpha^{14}\} & c_2 = \alpha^4 \quad b_1 = \alpha^7 \\ A_3 = \{\alpha^8, \alpha, \alpha^7\} & c_3 = \alpha \quad b_3 = \alpha^6 \\ A_4 = \{\alpha^9, \alpha^2, \alpha^8, \alpha^6\} & c_4 = \alpha^7 \quad b_4 = \alpha^3 \\ A_5 = \{\alpha^{16}, \alpha^9, \alpha^{15}, \alpha^{13}, \alpha^3\} & \end{array} \qquad b_0 = \alpha^3$$

Then $|A_5| = 5$ gives $wt(g(x)) \geqslant 5$. Also every

$c(x) = g(x)a(x) \in C$ contains S in its set of roots.

Computing the bound - example cont. II

Need to analyze the weight of any codeword !

Example (cont.)

- First note that any codeword can be written as

$$c(x) = g(x)a(x)$$

for some $a(x) \in \mathbb{F}[x]/x^n - 1$.

- Meaning that any codeword must contain the roots of g.

Computing the bound - example cont. III

Example (cont.)

- If $c(x)$ contains a root from C_3 then $m_3(x)|c(x)$ because $m_3(x)$ is defined as a polynomial

$$m_3(x) = \prod_{i \in C_3} (x - \alpha^i).$$

- Any root of $c(x)$ from C_3 implies that all the roots from C_3 are the roots of $c(x)$.

Computing the bound - example cont. IV

Example (cont.)

Need to consider other cases.

- The case $c(x) = (x-1)m_1(x)m_3(x) \equiv 0$ is easy $\mathbf{0}$ codeword.
- If $c(x) = m_1(x)m_3(x)$ then $wt(c) = 17$ as

$$\frac{x^n - 1}{x - 1} = 1 + x + x^2 + \cdots + x^{16}.$$

- In all cases $wt(c(x)) \geqslant 5$ for $c(x) \neq 0$ - that is $d \geqslant 5$.
- Note that always $A_i \setminus \{b_i\} \subset S$ for any A_i in the sequence.

Try to construct a sequence $(A_i : i \geqslant 0)$ of weight > 5 !

Summary

- Several bounds on the minimum distance were derived
- Might not be accurate but still sufficiently good for the estimation of the code's properties.
- Need to analyze decoding algorithms for BCH codes - efficiency
- Other interesting classes of codes may be derived from BCH codes

Chapter 5

Decoding BCH codes and Reed-Solomon codes

Contents of the chapter:

- Decoding BCH
- PGZ decoder
- Finding roots
- Reed-Solomon codes

BCH codes - reminder

- Recall that BCH codes with designated distance δ are defined through the set of consecutive roots $\alpha^a, \alpha^{a+1} \ldots, \alpha^{a+\delta-2}$ with,

$$g(x) = lcm\{m_i(x) : a \leqslant i \leqslant a+\delta-2\}$$

- Since BCH codes are linear one decoding alternative is to use standard array and syndrome decoding
- More efficient decoding is based on error locators and syndrome polynomial.

Parity check matrix - example

Example

Consider $g(x) = 1 + x + x^3$ that generates a binary $(7, 4)$ code. All the roots of $g(x)$ lie in $GF(2^3)$.

If α is a primitive element of $GF(2^3)$ generated by $x^3 + x + 1$, then α, α^2 and α^4 are roots of g.

But these roots are conjugates of each other. Therefore,

$$H = [1\ \alpha\ \alpha^2\ \alpha^3\ \alpha^4\ \alpha^5\ \alpha^6] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Designing BCH codes

Algorithm

- For given n the roots of

$$g(x) = \text{lcm}\{m_{\alpha_1}(x), \ldots, m_{\alpha_r}(x)\}$$

are chosen such that:

1. We get as much as possible consecutive roots of $m_i(x)$
2. For the least degree of g (increasing the dimension)

Decoding BCH codes - big picture

Idea

- Idea is to reconstruct the error polynomial $e(x)$ from the syndromes,

$$\begin{array}{ccccc} r(x) = c(x) + e(x) & \longrightarrow & S_i = r(\alpha^i) = e(\alpha^i) \\ & & \downarrow \\ c(x) & \longleftarrow & e(x) \end{array}$$

The main step of recovering $e(x)$ requires applying:

- The Extended Euclidean Algorithm (or PGZ) and
- Finding the roots of polynomials.

Decoding narrow-sense BCH codes

- Narrow-sense means $a = 1$, so the roots are $\alpha, \alpha^2, \ldots, \alpha^{\delta-1}$.
- As usual, $c(x)$ is a code polynomial and $e(x)$ is error polynomial with

$$wt(e(x)) = l \leqslant t = \lfloor (\delta - 1)/2 \rfloor.$$

- The first step is to compute the syndrome $s(x)$ of $r(x) = c(x) + e(x)$,

$$r(x) = h(x)g(x) + s(x),$$

by dividing $r(x)$ with $g(x)$ (EA).

- Of course, if $e(x) = g(x)p(x)$ or $e(x) = 0$ then $s(x) = 0$.

Decoding narrow-sense BCH codes II

Definition

Define

$$S_i = s(\alpha^{i+1}), \quad 0 \leqslant i \leqslant \delta - 2$$

evaluations of the syndrome at the roots of g.

- Then we have

$$s(\alpha^{i+1}) = r(\alpha^{i+1}).$$

Furthermore, $r(x) = c(x) + e(x)$ implies,

$$S_i = r(\alpha^{i+1}) = e(\alpha^{i+1}), \quad 0 \leqslant i \leqslant \delta - 2.$$

Error locators and evaluators

- Then let,

$$e(x) = \sum_{j=0}^{l-1} \lambda_{a_j} x^{a_j}, \quad a_j \in [0, n-1], \ \lambda_{a_j} \neq 0.$$

  Introduce

$$X_j = \alpha^{a_j}, \quad 0 \leqslant j \leqslant l-1$$

  (in the book u_j is used):

- The X_j are called error locators
- The a_j are called error location numbers
- The λ_{a_j} are called error magnitudes

Error locations and magnitudes

Just change of variables:

- error locators: $X_0 = \alpha^{a_0}, \ldots, X_{l-1} = \alpha^{a_{l-1}}$.
- error magnitudes: $Y_0 = \lambda_{a_0}, \ldots, Y_{l-1} = \lambda_{a_{l-1}}$

– Error locators are elements of the decoder alphabet $GF(q^m)$.

– Error magnitudes are elements of the channel alphabet $GF(q)$.

– Important special case: $Y_i = 1$ for channel alphabet $GF(2)$.

Our goal is to find error locators and magnitudes !

Syndrome equations

Partial syndromes

$$S_i = e(\alpha^{i+1}) = \sum_{j=0}^{l-1} Y_j X_j^{i+1},\ i = 0, \ldots, \delta - 2$$

are constants in system of $\delta - 1 = 2t$ equations in $2l$ unknowns

$$\begin{aligned} S_0 &= Y_0 X_0 + \ldots + Y_{l-1} X_{l-1} \\ S_1 &= Y_0 X_0^2 + \ldots + Y_{l-1} X_{l-1}^2 \\ &\vdots \\ S_{2t-1} &= Y_0 X_0^{\delta-1} + \ldots + Y_{l-1} X_{l-1}^{\delta-1} \end{aligned}$$

Error locator polynomial

Definition

The error-locator polynomial $\Lambda(x)$ is defined by

$$\begin{aligned} \Lambda(x) &= (1 - xX_0)(1 - xX_1) \cdots (1 - xX_{l-1}) \\ &= \prod_{j=0}^{l-1} (1 - xX_j) \\ &= 1 + \Lambda_0 x + \ldots + \Lambda_{l-1} x^l. \end{aligned}$$

The zeroes of $\Lambda(x)$ are $X_0^{-1}, \ldots, X_{l-1}^{-1}$ the reciprocals of error locators.

The degree of $\Lambda(x)$ is the number of errors. The decoder must determine l as well as the error locations.

Syndrome equations - solving

Need to solve an algebraic system of equations of degree $\delta - 1 = 2t$.

Goal: Using error locator polynomial we reduced the problem to one-variable polynomial equation with l solutions.

Before we give a formal approach let us consider an example of binary channel alphabet and 2EC BCH code.

Example - decoding 2EC BCH code

Example

Let $GF(2^4) = GF(2)[x]/(p(x))$, where

$$p(x) = x^4 + x + 1$$

is the primitive polynomial, and let α be the primitive root $\alpha^4 = \alpha + 1$.

We let $g(x)$ be defined through the zeroes:

$$\alpha, \alpha^2, \alpha^3 \alpha^4$$

and let $n = 15$ be the length of the code.

Example - decoding 2EC BCH code II

Example

- The minimum distance is $d \geqslant 5 = 2t + 1$; the BCH code C can correct $t = 2$ errors.
- $\alpha, \alpha^2, \alpha^4$ are all conjugates to each other (have the same minimum polynomial)
- α and α^3 are not conjugates
- $g(x)$ is simply the polynomial of smallest degree having α and α^3 as roots.

Example - decoding 2EC BCH code III

Example

Thus, the parity matrix is given by (check the lecture notes),

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{13} & \alpha^{14} \\ 1 & (\alpha^3)^1 & (\alpha^3)^2 & \cdots & (\alpha^3)^{13} & (\alpha^3)^{14} \end{pmatrix}$$

which simplifies to (using $\alpha^{15} = 1$),

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^9 & \alpha^{12} \end{pmatrix}$$

Example - decoding 2EC BCH code IV

Example

Assume $\mathbf{c}$ is sent over a BSC, and the received vector is

$$\mathbf{r} = \mathbf{c} + \mathbf{e}.$$

Further assume that exactly two errors have occured, i.e.,

$$\mathbf{e} = \mathbf{e}_i + \mathbf{e}_j$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are nonzero at positions i and j.

Channel alphabet is binary, but the decoder alphabet is GF(16) !!

Now we compute the syndrome of $\mathbf{r}$ as usual,

$$S = H\mathbf{r}^T = (\alpha^i + \alpha^j, (\alpha^i)^3 + (\alpha^j)^3) = (s_0, s_1).$$

Example - decoding 2EC BCH code V

Example

- α^i and α^j error locators, since their logs are the locations of the two errors (**no magnitude of error needed**).
- We construct the **error location polynomial** $S(x)$ from s_0 and s_1, which is a polynomial over GF(16) whose roots are the error locators

$$S(x) = (x + \alpha^i)(x + \alpha^j) = x^2 + (\alpha^i + \alpha^j)x + \alpha^{i+j}.$$

Then since, $\alpha^i + \alpha^j = s_0$ and $(\alpha^i)^3 + (\alpha^j)^3 = s_1$ we have,

$$s_1 = (\alpha^i + \alpha^j)[(\alpha^i)^2 + \alpha^{i+j} + (\alpha^j)^2] = s_0(s_0^2 + \alpha^{i+j}).$$

Example - decoding 2EC BCH code VI

Example

Using $\alpha^{i+j} = \frac{s_1}{s_0} + s_0^2$ we have,

$$S(x) = x^2 + (\alpha^i + \alpha^j)x + \alpha^{i+j} = x^2 + s_0 x + \frac{s_1}{s_0} + s_0^2$$

and we need to find the roots of this polynomial to get α^i and α^j !

Note that s_0, s_1 are known - the error correcting scheme is as:

- If $s_0 = s_1 = 0$, then we decide that no error has occurred.
- If $s_0 \neq 0$ and $s_1 = s_0^3$, then a single error at $z = \alpha^i$.
- If $s_0 \neq 0$ and $s_1 \neq s_0^3$, two errors have occurred: need to find α^i and α^j by finding the two roots of the $S(x)$.

Peterson-Gorenstein-Zierler decoder

$\Lambda(x)$ can be found from S_j using P-G-Z algorithm.

Example

Syndromes for 2EC narrow-sense BCH code with decoder alphabet $GF(2^m)$ are $S_j = Y_1 X_1^j + Y_2 X_2^j, \ \ j = 1, \ldots, 4$.

If 2 errors then zeroes of $\Lambda(x) = 1 + \Lambda_1 x + \Lambda_2 x^2$ are X_1^{-1} and X_2^{-1}

$$0 = 1 + \Lambda_1 X_1^{-1} + \Lambda_2 X_1^{-1} \overset{\cdot Y_1 X_1^3}{\rightarrow} Y_1 X_1^3 + \Lambda_1 Y_1 X_1^2 + \Lambda_2 Y_1 X_1 = 0$$

$$0 = 1 + \Lambda_1 X_2^{-1} + \Lambda_2 X_2^{-1} \overset{\cdot Y_2 X_2^3}{\rightarrow} Y_2 X_2^3 + \Lambda_1 Y_2 X_2^2 + \Lambda_2 Y_2 X_2 = 0$$

$$\underbrace{(Y_1 X_1^3 + Y_2 X_2^3)}_{S_3} + \Lambda_1 \underbrace{(Y_1 X_1^2 + Y_2 X_2^2)}_{S_2} + \Lambda_2 \underbrace{(Y_1 X_1 + Y_2 X_2)}_{S_1} = 0.$$

PGZ decoder II

Multiplying by $Y_i X^4$ and summing gives another equation:

$$\underbrace{(Y_1 X_1^4 + Y_2 X_2^4)}_{S_4} + \Lambda_1 \underbrace{(Y_1 X_1^3 + Y_2 X_2^3)}_{S_3} + \Lambda_2 \underbrace{(Y_1 X_1^2 + Y_2 X_2^2)}_{S_2} = 0.$$

Two linear equations in the unknowns Λ_1, Λ_2:

$$\begin{array}{l} S_3 + S_2\Lambda_1 + S_1\Lambda_2 = 0 \\ S_4 + S_3\Lambda_1 + S_2\Lambda_2 = 0 \end{array} \Rightarrow \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = - \begin{bmatrix} S_3 \\ S_4 \end{bmatrix}$$

The determinant of the coefficient matrix is:

$$S_1 S_3 - S_2^2 = Y_1 Y_2 (X_1 X_2^3 + X_1^3 X_2) = Y_1 Y_2 X_1 X_2 (X_1 + X_2)^2 \neq 0,$$

because $Y_i, X_i \neq 0$ and $X_1 \neq X_2$. So we can solve for Λ_1, Λ_2.

PGZ decoder III

Given Λ_1, Λ_2 need to find the roots of $\Lambda(x)$, i.e. X_1^{-1} and X_2^{-1} !

Use exhaustive search if no special structure of polynomial.

Remains to determine error magnitudes:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} X_1 & X_2 \\ X_1^2 & X_2^2 \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

The error magnitudes are given by (check that determinant $\neq 0$!):

$$\begin{aligned} Y_1 &= \frac{X_2 S_1 + S_2}{X_1(X_1 + X_2)} \\ Y_2 &= \frac{X_1 S_1 + S_2}{X_2(X_1 + X_2)} \end{aligned}$$

PGZ decoder - general view

By definition of the error locator polynomial, $\Lambda(X_i^{-1}) = 0$:

$$1 + \Lambda_1 X_i^{-1} + \Lambda_2 X_i^{-2} + \ldots + \Lambda_l X_i^{-l} = 0, \quad i = 1, \ldots, l \leqslant t.$$

Multiplying this equation by $Y_i X_i^{j+l}$ for any $j \geqslant 1$:

$$Y_i X_i^{j+l} + \Lambda_1 Y_i X_i^{j+l-1} + \ldots + \Lambda_l Y_i X_i^{j} = 0.$$

This equation has only positive powers of X_i . Now sum over i:

$$\sum_{i=1}^{l} Y_i X_i^{j+l} + \Lambda_1 \sum_{i=1}^{l} Y_i X_i^{j+l-1} + \ldots + \Lambda_l \sum_{i=1}^{l} Y_i X_i^{j} = 0.$$

PGZ decoder - general view II

Thus for $1 \leqslant j \leqslant 2t - l$ we may write,

$$S_{j+l} + \Lambda_1 S_{j+l-1} + \Lambda_2 S_{j+l-2} + \ldots + \Lambda_l S_j = 0.$$

There are $2t - l \geqslant l$ linear equations in l unknowns $\Lambda_1, \ldots, \Lambda_l$!

- The rest is easy: solve the system, find the roots of error locator polynomial and finally find error magnitudes.
- EEA method (the textbook) is quite similar in performance
- To me PGZ is the most elegant algorithm (if not the most efficient one)

Decoding - summary

- Need to find the roots of error locator polynomial.
- If the alphabet is $GF(2)$ then no need to find error magnitudes just locations.
- Finding magnitudes corresponds to finding a solution to a system of linear equations (see Example 16).
- Nevertheless, finding roots exhaustively can be demanding for large fields.

Finding roots of affine polynomials

Example

Find the roots of $l(x) = 1 + x^2 + x^8$ in $\mathbb{F} = GF(2^4)$. Let $\mathbb{F}$ be generated by α where $f(x) = 1 + x + x^4$, $f(\alpha) = 0$.

The elements of $\mathbb{F}$ are 4-tuples over $GF(2)$ with respect to the basis

$$B = \{1, \alpha, \alpha^2, \alpha^3\}.$$

If $\rho = (a_0, a_1, a_2, a_3)$ is a root of l then,

$$1 + (\sum_{i=0}^{3} a_i \alpha^i)^2 + (\sum_{i=0}^{3} a_i \alpha^i)^8 = 0.$$

Identify the free terms with 0, the terms that are coefficients of α to 0 etc. Get 4 linear equations and solve.

Finding roots in the field II

- Affine polynomial over the field $GF(p^n)$ is defined as,

$$l(x) = \lambda + \sum_{i=0}^{n-1} \lambda_i x^{p^i}.$$

- If we consider a cubic polynomial $h(x) = ax^3 + bx^2 + cx + d$ we can use a change of variables

$$x = y + b/a \rightarrow \tilde{h}(y) = ay^3 + (\frac{b^2}{a} + c)y + (\frac{cb}{a} + d).$$

- Then $y\tilde{h}(y)$ is a linearized polynomial ! The nonzero roots of $y\tilde{h}(y)$ are zeros of $\tilde{h}(y)$ and these gives the roots of $h(x)$.

Reed-Solomon codes - introduction

- Just a special class of BCH codes but important from application point of view
- Often used in applications that need a great capability of correcting burst errors
- Efficient encoding and decoding is possible.
- The main problem (asymptotically) is the alphabet size !

RS codes - discussion

Facts

So far we were analyzing cyclic codes specified by:

– Alphabet size (channel): q symbols from $GF(q)$ (q some prime power)

– Decoder alphabet :q^m our roots were in some $GF(q^m)$.

Reed-Solomon codes are defined so that the channel alphabet = decoder alphabet

RS codes - discussion II

How do we do that ?

- Simply we need an n-th primitive root of unity in $\mathbb{F} = GF(q)$. That is, some $\beta \in \mathbb{F}$ such that $ord(\beta) = n$ ($\beta^n = 1$)
- Since the code length is n the roots of our generator polynomial are found in the field $GF(q)$ not in some $GF(q^m)$!

Generator polynomial for RS codes

For some n-th primitive root of unity β define,

$$g(x) = (x - \beta^{1+a})(x - \beta^{2+a}) \cdots (x - \beta^{\delta-1+a})$$

for some $\delta \geqslant 2$ and some $a \geqslant 0$. Then,

- Clearly $g(x)|x^n - 1$ as $x^n - 1 = \prod_{i=0}^{n-1}(x - \beta^i)$
- It means that $g(x)$ generates an ideal in $\mathbb{F}/(x^n - 1)$ - cyclic code in $V_n(F)$.

RS code example

Example

Consider $GF(2^4)$ generated by a root of $f(x) = x^4 + x + 1$.

The element $\beta = \alpha^3$ is clearly a 5-th root of unity, i.e. $\beta^5 = 1$. Let,

$$g(x) = (x - \beta)(x - \beta^2)(x - \beta^3) = \alpha^3 + \alpha^2 x + \alpha^{11} x^2 + x^3$$

Then g generates a (5,2)-RS code over $GF(2^4)$ with designed distance $\delta = 4$.

The generator matrix of C is

$$G = \begin{bmatrix} \alpha^3 & \alpha^2 & \alpha^{11} & 1 & 0 \\ 0 & \alpha^3 & \alpha^2 & \alpha^{11} & 1 \end{bmatrix}$$

RS code example cont.

Example (Cont.)

Note that C has 256 codewords as the message space is $m(x) = a_0 + a_1 x$ where $a_0, a_1 \in GF(2^4)$.

The parity check matrix for this code is,

$$H = \begin{bmatrix} 1 & 0 & 0 & \alpha^3 & \alpha^{14} \\ 0 & 1 & 0 & \alpha^2 & \alpha^8 \\ 0 & 0 & 1 & \alpha^{11} & \alpha^{12} \end{bmatrix}$$

Check that H is a parity check matrix, and that $d = 4$ (HW) !

The error correcting and detection capability are given by,

- An (n, k)-RS code can detect $n - k$ errors or correct $\lfloor (n - k)/2 \rfloor$ errors.

Properties of RS codes

- Remark that $\beta^i \in \mathbb{F}$ for all i so the minimal polynomial of β^i is simply $(x - \beta^i)$!
- As degree of g is $\delta - 1$ this linear cyclic code has parameters (n, k) where $n - k = \delta - 1$.
- This is the definition of (n, k)-RS code !

Properties of RS codes II

- The RS code was defined through $\delta - 1$ consecutive roots. Then the BCH bound implies that $d \geqslant \delta$
- Can d be grater than δ ?
- NO, recall the Singelton bound $n - k \geqslant d - 1$, therefore as $n - k = \delta - 1$ by construction, we have $d = \delta$.

Facts

- Therefore an (n, k)-RS code is a maximum-distance separable (MDS) code !
- It means that for a given n and k no linear code can have larger distance than RS code !

RS codes over $GF(2^m)$

Two reasons for investigating RS codes over $GF(2^m)$

- The burst error correcting capabilities
- Ease of implementation over fields of characteristic 2

Instead of viewing an (n, k)-RS code C over $GF(2^m)$ we can consider associated code C' which is an (nm, km) code over $GF(2)$.

Replacement:

- Each coordinate $c_i \in GF(2^m)$ of $c \in C$ is replaced by $c_i' \in GF(2)^m$, i.e. a binary m-tuple.
- Then C' contains 2^{km} vectors over $GF(2)$ - a vector space of dimension km.

RS codes - burst error correction

What is the ability of C' to correct burst error $\mathbf{e}$ of length b ?

The code C can correct up to $\lfloor (n-k)/2 \rfloor$ errors.

This means that in C' a burst error of length b can at most influence $\lfloor (n-k)/2 \rfloor$ errors.

$$c = (0, 1, \alpha), \quad c_i \in GF(2^3) \leftrightarrow c' = (00\underbrace{0, 100}_{b \text{ errors}}, 010).$$

Therefore, the maximum length of burst error that can be corrected is,

$$b = m(\lfloor (n-k)/2 \rfloor - 1) + 1.$$

Need to have $\lfloor (n-k)/2 \rfloor - 1 > 0$ to demonstrate this. Previous example is not good as $n-k=3$ so that $\lfloor (n-k)/2 \rfloor - 1 = 0$.

Burst error correction - example

Example

Let again consider $GF(2^4)$ generated by a root of $f(x) = x^4 + x + 1$.

Taking $\beta = \alpha^3$ define,

$$g(x) = (x-\beta)(x-\beta^2)(x-\beta^3)(x-\beta^4)$$

Then g generates a (5,1)-RS code over $GF(2^4)$ with designed distance $\delta = 5$.

The generator matrix of the code is simply $G = [11111]$.

Since $\delta = 5$ the code C over $GF(2^4)$ is capable of correcting 2 errors. In this case $b = 5$ so the burst error correcting capability is much larger.

Decoding RS codes

Decoding RS codes same as BCH codes- using the P-G-Z algorithm we would do :

1. Compute the syndromes $S_1, \ldots, S_{\delta-1}$
2. Solve a linear system of equations retrieving $\Lambda_1, \ldots, \Lambda_l$
3. Find the roots of $\Lambda(x)$ thus getting error locations
4. Compute the error magnitudes by solving a linear system for $Y_1, \ldots, Y_l$.

Also some other more advanced techniques are known such as list decoding (not included in this course)

Summary

– Decoding of BCH (RS codes) can be done efficiently

– These techniques are subjected to the state-of-the-art improvements

– Very interesting class of codes is Reed-Solomon codes; not that good for random errors but excellent for burst error correction

– Appealing for applications where usually interleaving is also used for greater performance (next lecture)

Chapter 6

Channel erasures and digital audio applications

Contents of the chapter:

- Channel erasures
- Interleaving
- Error correction in CD

RS codes - reminder

- Good burst error-correcting capability of RS codes - due to large alphabet and MDS property.
- But we want to deal with channel erasures as well - special type of errors (known location) !
- Need to modify the decoding procedure - correct both errors and erasures
- For practical applications better techniques exist such as interleaving.

Channel erasures - definition

Definition

A *channel erasure* is an error with known position but unknown magnitude (cannot decide what was sent).

Usually as a result of : insufficient information at the decoder side, or simply information that the value is unreliable.

- The task of the decoder is to restore or "fill" the erasure positions.

The decoder works on the following data (codeword)

$$\mathbf{r} = (\_\ \alpha^6 \_\ \_\ 1)$$

Correctable erasures

– C an $[n, M]$ code over an alphabet A.

A received vector $\mathbf{r}$ having l erasures (and **no errors**) is correctable to $\mathbf{c}$ if:

$\mathbf{c}$ is unique among $|A|^l$ vectors agreeing with $\mathbf{r}$ in $n - l$ non-erasure positions.

$$\mathbf{r} = (\underbrace{c_1, c_2, \ldots, c_{n-l}}_{n-l \text{ correct}}, \underbrace{\_ \cdots \_}_{l \text{ erasures}})$$

A code C can correct u erasures if any received vector with $l \leqslant u$ erasures is correctable.

Channel erasures - main theorem

Theorem
An $[n, M]$ code over A with distance d. Then it is capable of correcting $d - 1$ erasures.

Proof.
Assume $\mathbf{r}$ has $l = d - 1$ erasures. Then,

– Among $|A|^l$ n-tuples that agree with $\mathbf{r}$ in the $n - l$ positions **at most one codeword** since $d(C) = d$.

– As there are only erasures (no errors) **exactly one codeword c** to which $\mathbf{r}$ can be corrected.

For linear codes erasures are determined by solving a linear system of equations □

Correcting erasures - example

Example

$GF(2^4)$ generated by α a root of $f(x) = x^4 + x + 1$. For $\beta = \alpha^3$ we defined

$$g(x) = (x - \beta)(x - \beta^2)(x - \beta^3) = \alpha^3 + \alpha^2 x + \alpha^{11} x^2 + x^3$$

Then g generates a (5,2)-RS code over $GF(2^4)$ with designed distance $\delta = 4$.

The generator matrix of C is

$$G = \begin{bmatrix} \alpha^3 & \alpha^2 & \alpha^{11} & 1 & 0 \\ 0 & \alpha^3 & \alpha^2 & \alpha^{11} & 1 \end{bmatrix}$$

What if we receive $\mathbf{r} = (\_\ \alpha^6 \_\ \_\ 1)$?

Correcting erasures - example cont.

Example (Cont.)

The decoder assigns 0's to erasure positions $\mathbf{r}_0 = (0\ \alpha^6\ 0\ 0\ 1)$,

$$\mathbf{r}_0 = \mathbf{c} + (u_1\ 0\ u_2\ u_3\ 0).$$

Since $H\mathbf{c}^T = 0$ we have that $H\mathbf{r}_0^T = H(u_1\ 0\ u_2\ u_3\ 0)^T$, i.e.,

$$\begin{bmatrix} 1 & 0 & 0 & \alpha^3 & \alpha^{14} \\ 0 & 1 & 0 & \alpha^2 & \alpha^8 \\ 0 & 0 & 1 & \alpha^{11} & \alpha^{12} \end{bmatrix} \mathbf{r}_0^T = \begin{bmatrix} \alpha^{14} \\ \alpha^{14} \\ \alpha^{12} \end{bmatrix} = \begin{bmatrix} u_1 + \alpha^3 u_3 \\ \alpha^2 u_3 \\ u_2 + \alpha^{11} u_3 \end{bmatrix}$$

Unique solution: $u_1 = \alpha^3$ $u_2 = \alpha^9$ $u_3 = \alpha^{12}$ so we get,

$$\mathbf{c} = \mathbf{r}_0 - (\alpha^3\ 0\ \alpha^9\ \alpha^{12}\ 0) = (\alpha^3\ \alpha^6\ \alpha^9\ \alpha^{12}\ 1).$$

Channels with erasures and errors

Some channels admit both errors and erasures. We need to both correct errors and fill the erasure positions.

Increased minimum distance of the code is required to deal with both type of "errors".

Theorem
An $[n, m]$ code C with minimum distance $d = 2t + u + 1$ can correct t errors and u erasures.

The proof uses the triangle inequality to show that there can not be two codewords to which r can be decoded if $d = 2t + u + 1$.

Channels with erasures and errors - binary case

– Suppose a **binary** linear (n, k)-code C with $d = 2t + u + 1$. Then,

- C can correct t errors and u erasures.
- If no erasures C will correct up to $t + \lfloor u/2 \rfloor$ errors.

What is the decoding procedure in case of both t errors and u erasures ?

1. Decoder forms two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$ as,

$$\mathbf{r}_0 = (\underbrace{r_1, \ldots, r_{n-l}}_{t \text{ errors}}, \underbrace{0 \cdots 0\ 0}_{l \leqslant u \text{ erasures}}) \quad \mathbf{r}_1 = (\underbrace{r_1, \ldots, r_{n-l}}_{t \text{ errors}}, \underbrace{1 \cdots 1\ 1}_{l \leqslant u \text{ erasures}})$$

2. At least one of $\mathbf{r}_0$ and $\mathbf{r}_1$ has distance at most $t + \lfloor u/2 \rfloor$ from the transmitted codeword.

Channels with erasures and errors - binary case II

To decode $\mathbf{r}$ the decoder uses some standard decoding technique:

- Standard array (syndrome) decoding
- Error-trapping decoding ...

– **Problem:** If both $\mathbf{r}_0$ and $\mathbf{r}_1$ are decoded to **different** codewords ?

– **Solution:** There will be exactly one codeword requiring at most t changes to non-erasure positions (the other codeword requires more than t changes)

Decoding binary linear code with erasures - example

Example

Consider some $(15, 7)$ binary linear code C with $d = 5$. Then C can correct 1 error and 2 erasures, or 2 errors with no erasures.

Suppose that $\mathbf{r} = (00001\ 00\_\ \_\ 0\ 00000)$ then,

$$\mathbf{r}_0 = (00001\ 00000\ 00000), \quad \mathbf{r}_1 = (00001\ 00110\ 00000)$$

Using error-trapping as the decoding procedure the decoded codewords are,

$$\mathbf{c}_0 = (00000\ 00000\ 00000), \quad \mathbf{c}_1 = (10001\ 01110\ 00000)$$

$\mathbf{c}_0$ differs from $\mathbf{r}$ in 1 non-erasure position; $\mathbf{c}_1$ in 2 – decode as $\mathbf{c}_0$.

Decoding linear code with erasures over $GF(q)$

For larger alphabets the decoding is more complex.

– Let $S = \{\mathbf{a} \in GF(q)^n : \mathbf{a} = \mathbf{r} \text{ at } n - l \text{ non-erasure positions } \}$.

– Suppose C is an (n, k) code over $GF(q)$ with $d = 2t + u + 1$, and $\mathbf{r}$ has at most $l \leqslant u$ erasures and $\leqslant t$ errors.

Algorithm

1. Select a vector $\mathbf{s} \in S$, note that $|S| = q^l$.
2. Determine the unique codeword $\mathbf{c}$ s.t. $d(\mathbf{c}, \mathbf{s}) \leqslant t + \lfloor u/2 \rfloor$, if any. Otherwise, select another $\mathbf{s} \in S$ and repeat step (2).
3. If $\mathbf{c}$ differs from $\mathbf{s}$ in at most t non-erasure positions decode $\mathbf{r}$ to $\mathbf{c}$, otherwise (1).

BCH Decoding with erasures and errors - example

Example

All the decoding details are given in the following example.

- A BCH code of length $n = 6$ over GF(7)
- Designed distance $\delta = 5$
- Generated by $g(x) = (x - \beta)(x - \beta^2)(x - \beta^3)(x - \beta^4)$, where β is n-th primitive root of unity, e.g. $\beta = 3$.
- C can correct up to t errors and u erasures if $\delta \geqslant 2t + u + 1$

Possible combinations for u, t are : $(t, u) = (0, 4), (1, 2), (2, 0)$.

BCH Decoding with erasures - example II

Example (Cont.)

Suppose $\mathbf{c} = (4\ 3\ 0\ 5\ 6\ 2)$ is transmitted codeword and

$$\mathbf{r} = (3\ 3\ \_\ 5\ \_\ 2)$$

is received – 1 error and 2 erasures.

Filling erasure positions with zeros gives,

$$\mathbf{r} = (3\ 3\ \mathbf{0}\ 5\ \mathbf{0}\ 2), \quad r_0(x) = 3 + 3x + 5x^2 + 2x^5$$

Then we compute $S_i = r_0(\beta^{i+1})$ for $i = 0, \ldots, 3$,

$$S_0 = r_0(3) = 3 \qquad S_1 = r_0(2) = 1$$
$$S_2 = r_0(6) = 0 \qquad S_3 = r_0(4) = 3$$

BCH Decoding with erasures - example II

Example (Cont.)

Main assumption $t = 1$ errors and $u = 2$ erasures.

Next step: Compute the syndrome and **erasure** locator polynomial.

$$S(x) = \sum_{i=0}^{\delta-2} S_i x^i = 3 + x + 3x^3$$

and the erasure positions 2 and 4 gives

$$\lambda(x) = (1 - \beta^2 x)(1 - \beta^4 x) = 1 + x + x^2.$$

Then one computes (where $\deg(T_1) \leqslant u - 1 = 1$),

$$\lambda(x)S(x) = 3 + 4x + 4x^2 + 4x^3 + 3x^4 + 3x^5 = T_1(x) + x^u T_2(x).$$

BCH Decoding with erasures - example III

Example (Cont.)

For $u = 2$ (number of erasures) we compute,

$$T_2(x) = 4 + 4x + 3x^2 + 3x^3.$$

The key equation for the channel with erasures is,

$$\hat{W}(x) \equiv \Lambda(x) T_2(x) \equiv \Lambda(x)(4 + 4x) \pmod{x^2},$$

GOAL: Find $\Lambda(x)$ and $\hat{W}(x)$ with degree restriction (important) satisfying above equation;$\deg(\hat{W}(x)) \leqslant t - 1$,$\deg(\Lambda(x)) \leqslant t$.

Apply EEA with $a(x) = x^2$ and $b(x) = 4 + 4x$ until the remainder polynomial r_i is of degree $< (\delta - u - 1)/2 = 1$.

BCH Decoding with erasures - example IV

Example (Cont.)

– EEA gives

$$\Lambda(x) = 5x + 2$$

as the error locator poly and

$$r_i = 1 = \hat{W}(x),$$

so $\beta^0 = 1$ is a root and

$$\beta^{0^{-1}} = \beta^0$$

gives the error location 0 !

Easy to check that

$$1 \equiv (5x + 2)(4 + 4x) \pmod{x^2}$$

BCH Decoding with erasures - example V

Example (Cont.)

Need to find magnitudes (treat both erasures and errors),

$$e(x) = Y_0 + Y_2 x^2 + Y_4 x^4$$

Using $S_i = e(\beta^{i+1}) = r_0(\beta^{i+1})$ we get,

$$\begin{aligned} S_0 &= e(3) = Y_0 + 2Y_2 + 4Y_4 = 3 \\ S_1 &= e(2) = Y_0 + 4Y_2 + 2Y_4 = 1 \\ S_2 &= e(6) = Y_0 + Y_2 + Y_4 = 0 \end{aligned}$$

A system over GF(7) with a solution $(Y_0, Y_2, Y_4) = (6, 0, 1)$.
Finally, $\mathbf{r} \to \mathbf{r}_0 - \mathbf{e} = (3\ 3\ 0\ 5\ 0\ 2) - (6\ 0\ 0\ 0\ 1\ 0) = (4\ 3\ 0\ 5\ 6\ 2)$.

Interleaving - introduction

Interleaving is a way of arranging data in a non-contiguous way to increase performance.

Typical usage:

- In error-correction coding, particularly within data transmission, disk storage, and computer memory.
- For multiplexing of several input data over shared media.
- In streaming media applications, it enables quasi-simultaneous reception of input streams, such as video and audio.
- For improved access performance in computer data storage.

Interleaving - example

Hard drives storages commonly divided into blocks (sectors).

Early computers was too slow to perform standard operations sequentially:

- Read a block of data into buffer
- Move the data somewhere else
- Be ready to read the next sector

Assume that a computer (to be ready for reading) needs the time corresponding to passage of 3 sectors

Then instead of sequential block ordering 123456789 one would use,

186429753

Codeword Interleaving

Main idea is to increase the burst error-correcting capabilities of a code.

We cannot construct a code (high redundancy) to correct long burst errors.

Instead of letting a single codeword correct a burst error we distribute the burst errors over many codewords.

The result is a random distribution of errors - error correcting capability can deal with these.

Codeword Interleaving - example

Example

Consider a single error-correcting (7,4) code C with,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Suppose we transmit,

$$\mathbf{c}_1 = (1100\ 001) \quad \mathbf{c}_2 = (0011\ 110) \quad \mathbf{c}_3 = (0111\ 000)$$

in this order.

Each codeword can correct a single error but none of them can correct a burst of length 2 !

Codeword Interleaving - example II

Example

What if we send the first bit of $\mathbf{c}_1$ then the 1st bit of $\mathbf{c}_2$ followed by the 1st bit of $\mathbf{c}_3$ etc.

$$\downarrow \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{matrix}$$

We send the data column-wise,

$$100\ \ 101\ \ 011\ \ 011\ \ 010\ \ 010\ \ 100$$

The 3 codewords are interleaved - any burst error of length at most 3 results in at most 1 error in each of the original codewords.

Single error-correcting (7,4) code into 3 burst EC (21,12) code.

Interleaving - generalization

– We consider an (n, k) code C that corrects any burst of length b.

Construct a new code C^* as follows:

For every collection of t codewords $(a_{i1}\ a_{i2} \ldots a_{in})$, $1 \leqslant i \leqslant t$ of C form a matrix,

$$T = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{t1} & a_{t2} & \cdots & a_{tn} \end{bmatrix}$$

The codewords of C^* are the columns of T.

C^* is an (nt, kt)-code which corrects bursts of length bt !

Cross-interleaving

Cross-interleaving is interleaving of several codes.

Important in practice:

- Compact disc (cross-interleaving of two Reed-Solomon codes)
- Interleaving of data packets (UDP) (many patents)
- Design of turbo codes etc.

The technique is quite cumbersome when more than two codes are involved.

IDEA: Instead of using the columns of interleaved codewords, these codewords are treated as information symbols of another code.

Cross-interleaving - example

Example

Previously we interleaved a (7,4) code with depth 3 by constructing,

$$\downarrow \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{matrix}$$

and reading column-wise.

Now we use another code, say a (6,3) (single-error correcting) code C_2 defined by,

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

and encode the first column 100 as 100111 (first column of G_2).

Cross-interleaving - example II

Example (Cont.)

The codewords of C_1

$$\mathbf{c}_1 = (1111\ 111) \quad \mathbf{c}_2 = (0111\ 000) \quad \mathbf{c}_3 = (0100\ 110)$$

gives,

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad \begin{matrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \\ 1 & 0 & 0 & 1 & 1 & 1 \end{matrix}$$

The codewords to the right can be interleaved to any desired depth (in the textbook depth is 2).

Cross-interleaving - example III

Example (Cont.)

Code with distance d can correct $d-1$ erasures but only $(d-1)/2$ errors. Decoding strategy :

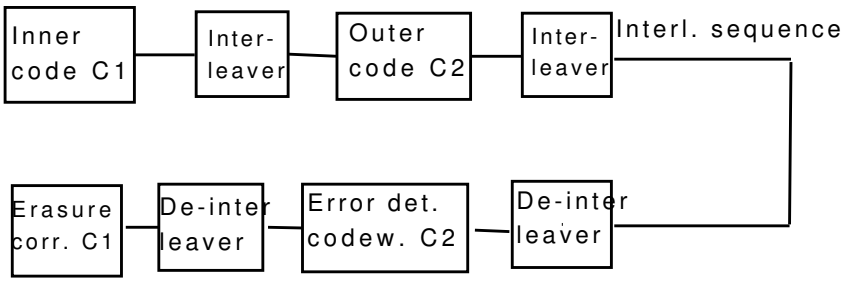
- Outer code C_2 used for error detection
- Inner code C_1 for (partial) error correction
- Interleaving of the inner codewords disperses errors (erasures) among inner codewords.

In the cross-interleaved string of length 36 (6 codewords of C_2) we can correct bursts of length 2.

But we use C_2 only for error detection (2 errors) and C_1 to correct erasures (2 erasures).

Cross-interleaving - example IV

Example (Cont.)



Suppose we detect errors in the two first codewords of C_2; then all information symbols are flagged as erasures e.g. 100 and 111.

De-interleaving results in at most 2 erasures in codewords of C_1,

$\mathbf{c}_1 = (\quad 11111) \quad \mathbf{c}_2 = (\quad 11000) \quad \mathbf{c}_3 = (\quad 00110)$

Delayed interleaving

Main idea is to efficiently process the data to be interleaved.

Instead of interleaving fixed blocks of n codewords, the codewords are interleaved in a continuous sequence.

- Codewords distributed diagonally over the interleaving array
- Using delay technique to obtain continuous stream
- Generalization from 1-frame delayed interleaver to d-frame delayed interleaver

Used in digital audio - 4-frame delay interleaver.

Digital audio

– For sound to be "CD quality" it must be sampled at 44.1 kHz at 16 bits by two channels.

– This gives a data rate of 1,411,200 bits/sec.

– 16 bits provides 65,536 (2^{16}) levels of sound information every $2.268x10^{-5}$ seconds.

– The audio data is broken into frames of 6 16bit words/channel.

– This data is then converted to 24 bytes words called frames.

– A "logical" choice for the inner code C_1 is a (28,24)-RS code over $GF(2^8)$.

Cross-Interleaved Reed-Solomon codes - CIRC

The inner code C_1 is interleaved to a depth of 28 using - symbols separated at distance 28 (can correct bursts of length 28).

– The outer code C_2 is a (32,28) RS code using the 28 symbols (over $GF(2^8)$) as its information symbols.

– Known as Cross-Interleaved Reed-Solomon codes (CIRC)

We focus on the burst error-correcting capability of CIRC.

The outer code C_2 has $d = 5$ - thus several mode of usage.

Cross-Interleaved Reed-Solomon codes II

Error detection/correction of C_2 ($n = 32$, $k = 28$, $q = 256$):

Error detection	Error correction
4	0
0	2
3	1

The probability that 4 or more errors going **undetected**:

$$\frac{q^k(1 + n(q-1))}{q^n} = 2^{-19}.$$

What if we use full error-correcting capability $t = 2$?

$$\frac{q^k(1 + n(q-1) + \binom{n}{2}(q-1)^2)}{q^n} = 2^{-7}.$$

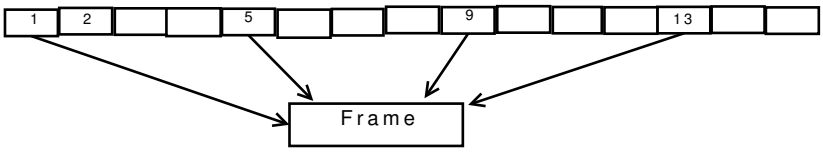
Decoding strategy

Algorithm

- If the decoder for C_2 detects a single error then it is corrected.
- Otherwise, all 28 information symbols are flagged as erasures ! The symbols are delay **de-interleaved** before reaching the decoder for C_1.
- The inner code C_1 has distance 5 as well - designed to correct 4 erasures.

Due to 4-frame interleaving the decoder for C_1 is capable of correcting 16-frame bursts$\approx 16 \times 24 \times 8 = 3000$ audio bits !

Interleaving effect



4 erasures decodable by C1 decoder

– Apart from error-correcting the CIRC code also uses concealing.

– If C_1 cannot correct some sample then its value can be "recovered" by interpolating with two neighboring reliable samples.

Summary

– Channel erasures is a very useful concept when treating burst errors

– CIRC error correction is one of the greatest applications of coding theory in the 20th century

– Google results on submitted patents with improvements on standard CIRC are numerous

– Design of interleaver is crucial for so-called turbo codes - the care must be taken regarding the size of the interleaver.

– Decoding the RS codes is also appealing patent issue: both complexity of decoding and in particular efficient computation in the finite field.