CO 331 - Coding Theory

Cameron Roopnarine

Last updated: April 15, 2021

Contents

List of Algorithms

Chapter 1

Introduction and Fundamentals

2020-01-06

1.1 An Introduction to Coding Theory

$$\text{Source} \to \underset{\text{(digital data)}}{\text{Source Encoder}} \to \underset{\text{(encoding algorithm)}}{\text{Channel Encoder}} \xrightarrow[\uparrow \text{Noise}]{\text{Channel}} \underset{\text{(decoding algorithm)}}{\text{Channel Decoder}} \to \text{Source Decoder} \to \text{Data}$$

EXAMPLE 1.1.1: Repetition Code

Message → Codeword	Errors/Codeword Detected	Errors/Codeword Corrected	Rate
$0 \to 0$ $1 \to 1$	0	0	1
$0 \to 00$ $1 \to 11$	1	0	$1/2$
$0 \to 000$ $1 \to 111$	2	1	$1/3$
$0 \to 00000$ $1 \to 11111$	4	2	$1/5$

Goal of Coding Theory

Design codes such that:

- High information rate
- High error-correcting capability
- Efficient encoding and decoding algorithms

Codes $\supset$ Block codes $\supset$ Linear codes $\supset$ Cyclic codes $\supset$ BCH Codes $\supset$ RS Codes

Codes not covered in this course:

- Flamming codes.

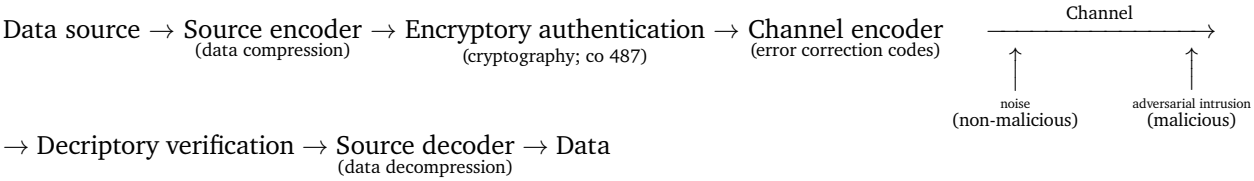
- Golay codes.
- Raptor codes.
- LDPC codes.
- Turbo codes.

Requirements for this course:

- MATH 136.
- Not required (but required to take the course): MATH 235.
- Familiarity with: Groups, Fields, Ideals, Rings (these will be taught)
- Useful, if you have completed these you might be bored: PMATH 336, PMATH 334 [or the advanced equivalents].

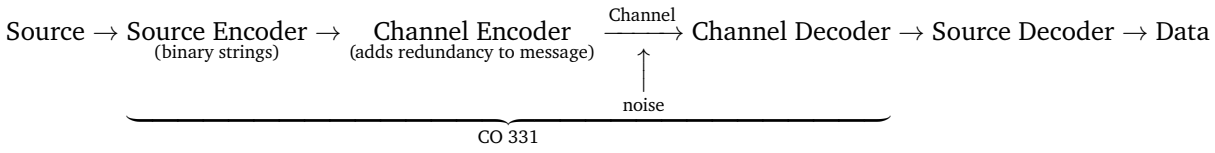
The Big Picture

In its broadest sense, coding deals with the reliable, efficient, and secure transmissions of data over channels that are subject to inadvertent noise and malicious intrusion.



2020-01-08

1.2 Fundamental Concepts



DEFINITION 1.2.1: Alphabet

An **alphabet** A is a finite set of $|A| = q \geqslant 2$ symbols.

DEFINITION 1.2.2: Word

A **word** is a finite sequence (**tuples** or **vectors**) of symbols from an alphabet A.

DEFINITION 1.2.3: Length

The **length** of a word is the number of symbols in it.

DEFINITION 1.2.4: Code

A **code** C over A is a finite set of words in A with $|C| \geqslant 2$.

DEFINITION 1.2.5: Codeword

A **codeword** $\boldsymbol{c}$ is a word in a code C.

DEFINITION 1.2.6: Block code

A **block code** is a code where all codewords have the same length. A block code C of length n containing M codewords over A is a subset $C \subseteq A^n$, with $|C| = M$. We refer to such a block code as an $[n, M]$-code over A.

EXAMPLE 1.2.7: Block Code

Let $A = \{0, 1\}$ and $C = \{00000,\ 11100,\ 00111,\ 10101\}$. C is a $[5, 4]$-code over $\{0, 1\}$.

Message	$\rightarrow$	Codeword
00	$\rightarrow$	00000
10	$\rightarrow$	11100
01	$\rightarrow$	00111
11	$\rightarrow$	10101

The encoding is a one-to-one map.

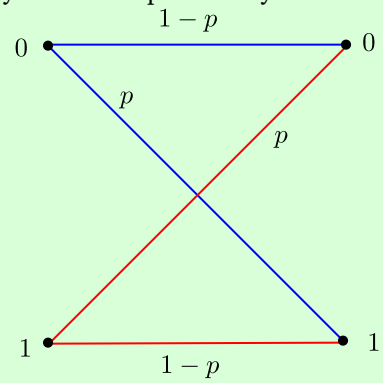
The channel encoder transmits only codewords, but what's received by the channel decoder might not be a codeword. For example, suppose the channel decoder receives $\boldsymbol{r} = 11001$. What should it do? In our above example, we can see that $\boldsymbol{r}$ is closest to 11100 and 10101 (only two bits are different), so it's possible that the codeword was one of those two. However, this may not be the case in practice.

DEFINITION 1.2.8: Assumptions about the Communications Channel

(I) The channel only transmits symbols from A.
(II) No symbols are deleted, added, or transposed.
(III) Errors are random.

EXAMPLE 1.2.9: Binary Symmetric Channel (BSC)

Let $A = \{0, 1\}$, and p denote the symbol error probability. The encoding map is:



A similar encoding map can be drawn for $A = \{0, 1, 2\}$, with symbol error probability $p/2$.
Suppose that the symbols transmitted are $X_1, X_2, ...$, and the symbols received are $Y_1, Y_2, ...$ Then for

all $i \geqslant 1$, $j \geqslant 1$, $k \leqslant q$, the probability that Y_i is received, given that X_i is transmitted is:

$$\mathbb{P}(Y_i = a_j \mid X_i = a_k) = \begin{cases} 1-p, & \text{if } j = k \\ \dfrac{p}{q-1}, & \text{if } j \neq k \end{cases}$$

DEFINITION 1.2.10: Notes about the Binary Symmetric Channel

(I) If $p = 0$, the channel is **perfect**.
(II) If $p = 1/2$, the channel is **useless**.
(III) If $1/2 < p \leqslant 1$, then simply flip all bits that are received.
(IV) WLOG, we can assume $0 < p < 1/2$.
(V) Analogously, for a q-ary channel, we can assume that $0 < p < \frac{q-1}{q}$.

DEFINITION 1.2.11: Hamming distance

If $\boldsymbol{x}, \boldsymbol{y} \in A^n$, the **Hamming distance** $d(\boldsymbol{x}, \boldsymbol{y})$ is the number of coordinate positions in which $\boldsymbol{x}$ and $\boldsymbol{y}$ differ.

EXAMPLE 1.2.12: Hamming Distance

Let $\boldsymbol{x} = 10111$ and $\boldsymbol{y} = 01010$. The Hamming distance of $\boldsymbol{x}$ and $\boldsymbol{y}$ is $d(\boldsymbol{x}, \boldsymbol{y}) = 4$ since $\boldsymbol{x}$ and $\boldsymbol{y}$ differ in the coordinate positions 1, 2, 3, and 5.

DEFINITION 1.2.13: Hamming distance

Let C be an $[n, M]$-code. The **Hamming distance d of a code C** is

$$d(C) = \min\{d(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x}, \boldsymbol{y} \in C,\ \boldsymbol{x} \neq \boldsymbol{y}\}$$

THEOREM 1.2.14: Metric

*d is a **metric**; that is, for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in A^n$:*
(1) $d(\boldsymbol{x}, \boldsymbol{y}) \geqslant 0$
(2) $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ *if and only if* $\boldsymbol{x} = \boldsymbol{y}$
(3) $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$
(4) (Triangle inequality): $d(\boldsymbol{x}, \boldsymbol{z}) \leqslant d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$

Proof of Theorem 1.2.14

Proof of (1) to (3) are trivially true.
Proof of (4) Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in A^n$. Suppose that $\boldsymbol{x}$ and $\boldsymbol{z}$ differ in exactly a positions; that is, $d(\boldsymbol{x}, \boldsymbol{z}) = a$. Out of the a positions in which $\boldsymbol{x}$ and $\boldsymbol{z}$ differ, there are b positions in which $\boldsymbol{y}$ is identical to $\boldsymbol{x}$, but not $\boldsymbol{z}$. Out of the a positions, there are $a - b$ positions in which $\boldsymbol{y}$ is identical to $\boldsymbol{z}$, but not $\boldsymbol{x}$. Lastly, in the $n - a$ positions where $\boldsymbol{x}$ is identical to $\boldsymbol{z}$, there are c positions in which $\boldsymbol{y}$ does not match either $\boldsymbol{x}$ or $\boldsymbol{z}$. We can see that $d(\boldsymbol{x}, \boldsymbol{y}) = b + C$ and $d(\boldsymbol{y}, \boldsymbol{z}) = a - b + C$. We get

$$d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z}) = (b + C) + (a - b + C) = a + 2c \geqslant a$$

Therefore d is a metric.

DEFINITION 1.2.15: Rate

The **rate** (or **information rate**) of an $[n, M]$-code C over A, is

$$R = \frac{\log_q(M)}{n}$$

where $q = |A|$.
If the source messages are all k-tuples over A, then $M = q^k$, so we have

$$R = \frac{\log_q(q^k)}{n} = \frac{k}{n}$$

EXAMPLE 1.2.16: Rate and Distance of a Code

Let $A = \{0, 1\}$ and $C = \{00000,\ 11100,\ 00111,\ 10101\}$ which is a $[5, 4]$-code over $\{0, 1\}$.

- The rate of C is $R = 2/5$.
- The distance of C is $d(C) = 2$, since the minimum distance are from the pair of codewords 00111 and 10101 which have Hamming distance of 2 as they differ in coordinate positions 1 and 4.

2020-01-10

1.3 Decoding Strategy

Suppose we have an $[n, M]$-code C over A of distance d. We need to adopt a strategy for the channel decoder (henceforth called the decoder). If decoder receives an n-tuple $\boldsymbol{r} \in A^n$, it must make some decision. This decision may be one of

(i) No errors have occurred; accept $\boldsymbol{r}$ as a codeword.

(ii) Errors have occurred; correct $\boldsymbol{r}$ to a codeword $\boldsymbol{c}$; e.g., $0 \to 0000$, $1 \to 1111$, $r = 0001$ corrected to 0000.

(iii) Errors have occurred; no correction is possible.

1.4 Nearest Neighbour Decoding

Incomplete Maximum Likelihood Decoding (IMLD)

Correct $\boldsymbol{r}$ to the unique codeword $\boldsymbol{c}$ for which $d(\boldsymbol{r}, \boldsymbol{c})$ is smallest. If $\boldsymbol{c}$ is not unique, reject $\boldsymbol{r}$.

Complete Maximum Likelihood Decoding (CMLD)

Same as IMLD, except ties are broken arbitrarily.

Question: Is IMLD a reasonable strategy?

THEOREM 1.4.1

IMLD selects the codeword $\boldsymbol{c}$ that maximizes $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c})$; that is, it maximizes the probability $\boldsymbol{r}$ is received, given $\boldsymbol{c}$ was sent.

We actually want to maximize $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c})$, but we will ignore that for now.

Proof of Theorem 1.4.1

Suppose $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ with $d(\boldsymbol{c}_1, \boldsymbol{r}) = d_1$ and $d(\boldsymbol{c}_2, \boldsymbol{r}) = d_2$. Suppose $d_1 > d_2$. Now, $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_1) = (1-p)^{n-d_1}\left(\frac{p}{q-1}\right)^{d_1}$ and $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_2) = (1-p)^{n-d_2}\left(\frac{p}{q-1}\right)^{d_2}$.
Hence,

$$\begin{aligned}\frac{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_1)}{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_2)} &= (1-p)^{d_2-d_1}\left(\frac{p}{q-1}\right)^{d_1-d_2} \\ &= \left[\frac{p}{(1-p)(q-1)}\right]^{d_1-d_2}\end{aligned}$$

Recall that, for a q-ary channel, we can assume that $p < \frac{q-1}{q}$. Thus,

$$\begin{aligned}&\implies pq < q-1 \\ &\implies 0 < q-1-pq \\ &\implies p < q-1-pq+p \\ &\implies p < (1-p)(q-1) \\ &\implies \frac{p}{(1-p)(q-1)} < 1\end{aligned}$$

Since $d_1 > d_2$, we get $\dfrac{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_1)}{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_2)} < 1$, and so $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_1) < \mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_2)$.

The ideal strategy is to correct $\boldsymbol{r}$ to $\boldsymbol{c} \in C$ such that $\mathbb{P}(\boldsymbol{c} \mid \boldsymbol{r})$ is maximized. This is **Minimum Error Decoding (MED)**.

EXAMPLE 1.4.2: IMLD $\neq$ MED

Let $C = \{\underbrace{000}_{\boldsymbol{c}_1}, \underbrace{111}_{\boldsymbol{c}_2}\}$, $\mathbb{P}(\boldsymbol{c}_1) = 0.1$, $\mathbb{P}(\boldsymbol{c}_2) = 0.9$, $p = 1/4$, and $\boldsymbol{r} = 100$.
IMLD $\boldsymbol{r}$ is decoded to $\boldsymbol{c}_1 = 000$.
MED

$$\begin{aligned}\mathbb{P}(\boldsymbol{c}_1 \mid \boldsymbol{r}) &= \frac{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_1)\mathbb{P}(\boldsymbol{c}_1)}{\mathbb{P}(\boldsymbol{r})} \\ &= \frac{p(1-p)^2(0.1)}{\mathbb{P}(\boldsymbol{r})} \\ &= \frac{0.0140625}{\mathbb{P}(\boldsymbol{r})}\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\boldsymbol{c}_2 \mid \boldsymbol{r}) &= \frac{\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_2)\mathbb{P}(\boldsymbol{c}_2)}{\mathbb{P}(\boldsymbol{r})} \\ &= \frac{p^2(1-p)(0.9)}{\mathbb{P}(\boldsymbol{r})} \\ &= \frac{0.0421875}{\mathbb{P}(\boldsymbol{r})}\end{aligned}$$

Since $\mathbb{P}(\boldsymbol{c}_1 \mid \boldsymbol{r}) < \mathbb{P}(\boldsymbol{c}_2 \mid \boldsymbol{r})$, $\boldsymbol{r}$ is decoded to $\boldsymbol{c}_2 = 111$.

Notes:

(i) IMLD selects $\boldsymbol{c}$ such that $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c})$ is maximum.

(ii) MED selects $\boldsymbol{c}$ such that $\mathbb{P}(\boldsymbol{c} \mid \boldsymbol{r})$ is maximum.

(iii) MED has a drawback that it requires knowledge of $\mathbb{P}(\boldsymbol{c}_i)$ for each $i \in [1, M]$.

(iv) Suppose source messages are equally likely, so $\mathbb{P}(\boldsymbol{c}_i) = \frac{1}{M}$ for each $i \in [1, M]$. Then,

$$\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_i) = \frac{\mathbb{P}(\boldsymbol{c}_i \mid \boldsymbol{r})\mathbb{P}(\boldsymbol{r})}{\mathbb{P}(\boldsymbol{c}_i)} = \mathbb{P}(\boldsymbol{c}_i \mid \boldsymbol{r}) \underbrace{M\mathbb{P}(\boldsymbol{r})}_{\text{constant}}$$

So, maximizing $\mathbb{P}(\boldsymbol{r} \mid \boldsymbol{c}_i)$ is the same as maximizing $\mathbb{P}(\boldsymbol{c}_i \mid \boldsymbol{r})$. Thus, IMLD is the same as MED in this case.

In the remainder of the course, we will use IMLD/CMLD.

2020-01-13

1.5 Error Correcting & Detecting Capabilities of a Code

- If C is used for error correction, the strategy is IMLD/CMLD.
- If C is used for error detection only, the strategy is to reject $\boldsymbol{r}$ if $\boldsymbol{r} \notin C$, otherwise accept $\boldsymbol{r}$.

DEFINITION 1.5.1: e-error correcting code

A code C is called an ***e*-error correcting code** if the decoder always makes the correct decision if at most e errors per codeword are introduced per transmission. We define ***e*-error detecting code** similarly.

EXAMPLE 1.5.2: Error Detecting and Correcting Codes

- $C = \{0000, 1111\}$ is a 1-error correcting code, but not a 2-error correcting code.
- $C = \{\underbrace{0\cdots0}_{m}, \underbrace{1\cdots1}_{m}\}$ is a $\lfloor \frac{m-1}{2} \rfloor$-error correcting code.
- $C = \{0000, 1111\}$ is a 3-error detecting code.

THEOREM 1.5.3

If $d(C) = d$, then C is a $(d-1)$-error detecting code.

Proof of Theorem 1.5.3

Suppose $\boldsymbol{c} \in C$ is transmitted $\boldsymbol{r}$ is received. If no errors occurred during transmission, then $\boldsymbol{r} = \boldsymbol{c}$, so the decoder correctly accepts $\boldsymbol{r}$. If at least 1 and at most $(d-1)$ errors occur, then $1 \leqslant d(\boldsymbol{r}, \boldsymbol{c}) \leqslant d-1$. Since $d(C) = d$, we have $\boldsymbol{r} \notin C$. Thus, the decoder correctly rejects $\boldsymbol{r}$. Thus, C is a $(d-1)$-error detecting code.

THEOREM 1.5.4

If $d(C) = d$, then C is not a d-error detecting code.

Proof of Theorem 1.5.4

Since $d(C) = d$, there exists codewords $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ with $d(\boldsymbol{c}_1, \boldsymbol{c}_2) = d$. Hence, it is possible that $\boldsymbol{c}_1$ is sent, d errors are introduced, and $\boldsymbol{c}_2$ is received. In this case, the decoder incorrectly accepts $\boldsymbol{c}_2$. Thus, C is not a d-error detecting code.

THEOREM 1.5.5

If $d(C) = d$, then C is a $\left\lfloor \frac{d-1}{2} \right\rfloor$-error correcting code.

Proof of Theorem 1.5.5

Suppose $\boldsymbol{c} \in C$ is transmitted, at most $\frac{d-1}{2}$ errors are introduced, and $\boldsymbol{r}$ is received. Let $\boldsymbol{z} \in C$ with $\boldsymbol{z} \neq \boldsymbol{c}$. By the triangle inequality, we have

$$\begin{aligned} d(\boldsymbol{c}, \boldsymbol{z}) \leqslant d(\boldsymbol{c}, \boldsymbol{r}) + d(\boldsymbol{r}, \boldsymbol{z}) \implies d(\boldsymbol{r}, \boldsymbol{z}) &\geqslant d(\boldsymbol{c}, \boldsymbol{z}) - d(\boldsymbol{c}, \boldsymbol{r}) \\ &\geqslant d - \frac{d-1}{2} \\ &= \frac{d+1}{2} \\ &> \frac{d-1}{2} \end{aligned}$$

So, $\boldsymbol{c}$ is the unique codeword closest to $\boldsymbol{r}$. Hence, IMLD/CMLD will decode $\boldsymbol{r}$ to $\boldsymbol{c}$. Thus, C is a $\left\lfloor \frac{d-1}{2} \right\rfloor$-error correcting code.

THEOREM 1.5.6

If $d(C) = d$, then C is not a $\left(\left\lfloor \frac{d-1}{2} \right\rfloor + 1 \right)$-error correcting code.

EXERCISE 1.5.7

Prove Theorem 1.5.6.

Given q, n, M, d, does there exist an $[n, M]$-code over A with $|A| = q$ such that $d(C) = d$?

Let $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M\}$ and $e = \lfloor \frac{d-1}{2} \rfloor$. For any codeword $\boldsymbol{c} \in C$, let $S_{\boldsymbol{c}}$ be the sphere of radius e centred at $\boldsymbol{c}$; that is,

$$S_{\boldsymbol{c}} = \{\boldsymbol{r} \in A^n : d(\boldsymbol{r}, \boldsymbol{c}) \leqslant e\}$$

We proved that if $\boldsymbol{c}_i, \boldsymbol{c}_j \in C$ with $i \neq j$, then $S_{\boldsymbol{c}_i} \cap S_{\boldsymbol{c}_j} = \emptyset$ for each $i \neq j$. This question can be viewed as a **sphere packing problem**: Can we place M spheres of radius e in A^n such that no two spheres overlap? This is a purely combinatorial problem.

Does there exist a block code with parameters $q = 2$, $n = 128$, $M = 2^{64}$, $d \geqslant 22$? Yes, we will see this in Chapter 6.

Road map

We'll view $\{0, 1\}^n$ as a vector space of dimension n over $\mathbb{Z}_q$ where $|A| = q$. We will choose the code C to be an M-dimensional subspace of this vector space, and we will choose special subspaces that satisfy the $d(C) = d$ requirement.

Chapter 2

Finite Fields

2020-01-15

2.1 Introduction

DEFINITION 2.1.1: Field

A **field** F is a set of elements under two binary operations, which we denote by $+$ and $\cdot$, such that $+ \colon F \times F \to F$ and $\cdot \colon F \times F \to F$ where all the following axioms are satisfied:

V1 $a + (b + c) = (a + b) + c$.
V2 $a + b = b + a$.
V3 $\exists\ 0 \in F$ such that $a + 0 = a$.
V4 $\exists\ (-a) \in F$ such that $a + (-a) = 0$.
V5 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
V6 $a \cdot b = b \cdot a$.
V7 $\exists\ 1 \in F$ such that $a \cdot 1 = a$.
V8 $\forall\ a \neq 0, \exists\ (a^{-1}) \in F$ such that $a \cdot (a^{-1}) = 1$.
V9 $a \cdot (b + c) = a \cdot b + a \cdot c$.

DEFINITION 2.1.2: Infinite Field

A field F is **infinite** if $|F|$ is infinite.

DEFINITION 2.1.3: Finite Field

A field F is **finite** if $|F|$ is finite.

DEFINITION 2.1.4: Order

The **order** of a field F denoted by $\text{ord}(F)$, is $|F|$.

EXAMPLE 2.1.5: Infinite and Finite Fields

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ are infinite fields.
- $\mathbb{Z}$ is **not** a field since $3 \in \mathbb{Z}$, but $(1/3) \notin \mathbb{Z}$.

Question: For what $n \in \mathbb{Z}_{\geqslant 2}$ does there exist finite fields of order n? If a field of order n exists, how do we "construct" it?

Recall: Let $n \in \mathbb{Z}_{\geqslant 2}$. The integers modulo n, denoted by $\mathbb{Z}_n$, is the set of all equivalence classes modulo n.

$$\mathbb{Z}_n = \{[0], [1], [2], \dots, [n-1]\}$$

where $[a]+[b] = [a+b]$ and $[a][b] = [ab]$. More simply, $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ with addition and multiplication performed modulo n.

EXAMPLE 2.1.6: Modular Arithmetic

Let $\mathbb{Z}_9 = \{0, 1, \dots, 8\}$.
- $5 + 7 = 3$; that is, $5 + 7 \equiv 3 \pmod 9$
- $5 \cdot 7 = 8$; that is, $5 + 7 \equiv 8 \pmod 9$

DEFINITION 2.1.7: Commutative ring

A **commutative ring** satisfies field axioms V1-V9 except V8.

THEOREM 2.1.8

$\mathbb{Z}_n$ is a commutative ring.

THEOREM 2.1.9

$\mathbb{Z}_n$ is a field if and only if n is prime.

Proof of Theorem 2.1.9

($\Longleftarrow$) Suppose n is prime. Let $a \in \mathbb{Z}_n$, $a \neq 0$ (i.e., $1 \leqslant a \leqslant n-1$). Since n is prime, $\gcd(a, n) = 1$ so $\exists\ s, t \in \mathbb{Z}$ such that

$$as + nt = 1$$

Reducing both sides modulo n gives

$$as \equiv 1 \pmod n$$

Define $a^{-1} = s$. Thus, V8 is satisfied and hence $\mathbb{Z}_n$ is a field of order n.
($\Longrightarrow$) Suppose for a contradiction that n is composite, say $n = ab$ where $2 \leqslant a, b \leqslant n-1$. Suppose a^{-1} exists, and define $a^{-1} = s$. Then,

$$as \equiv 1 \pmod n \implies abs \equiv b \pmod n \implies ns \equiv b \pmod n \implies 0 \equiv b \pmod n$$

So, $n \mid b$ which is impossible. Therefore, a^{-1} does not exist, and hence $\mathbb{Z}_n$ is not a field.

Question: Do there exist finite fields of orders 4 and 6?

DEFINITION 2.1.10: Characteristic

The **characteristic** of a field denoted by $\text{char}(F)$, is the smallest positive integer m such that

$$\underbrace{1 + \cdots + 1}_{m} = 0$$

If no such m exists, then we define $\text{char}(F) = 0$

EXAMPLE 2.1.11: Characteristic of Fields

- $\text{char}(\mathbb{Q}) = 0$
- $\text{char}(\mathbb{R}) = 0$
- $\text{char}(\mathbb{C}) = 0$

- $\text{char}(\mathbb{Z}_p) = p$ where p is prime.

THEOREM 2.1.12

If $\text{char}(F) = 0$*, then* F *is infinite.*

Proof of Theorem 2.1.12

Consider $1, 1+1, \ldots, \underbrace{1+\cdots+1}_{a} \in F$. Suppose for a contradiction there exists distinct $a, b \in \mathbb{Z}$ such that

$$\underbrace{1+\cdots+1}_{a} = \underbrace{1+\cdots+1}_{b}$$

where $a > b$, then

$$\underbrace{1+\cdots+1}_{a} = \underbrace{1+\cdots+1}_{b} + \underbrace{1+\cdots+1}_{a-b} = \underbrace{1+\cdots+1}_{b}$$

Hence, $\underbrace{1+\cdots+1}_{a-b} = 0 \implies \text{char}(F) = (a-b)$ which contradicts $\text{char}(F) = 0$. Thus, F is infinite.

THEOREM 2.1.13

If F *is a finite field, then* $\text{char}(F)$ *is prime.*

Proof of Theorem 2.1.13

Suppose for a contradiction that $\text{char}(F) = m$ is composite, say $m = ab$ where $2 \leqslant a, b \leqslant m-1$. Now

$$(\underbrace{1+\cdots+1}_{a})(\underbrace{1+\cdots+1}_{b}) = \underbrace{1+\cdots+1}_{m} = 0$$

since $\text{char}(F) = m$. Let $\underbrace{1+\cdots+1}_{a} = s$ and $\underbrace{1+\cdots+1}_{b} = t$, so $st = 0$ where $s \neq 0$. Since $\text{char}(F) = m > a$, there exists $c \in F$ such that $cs = 1 \implies c = s^{-1}$. Therefore, $s^{-1}st = 0$. Thus, $t = 0$ which is a contradiction to $\text{char}(F) = m$.

Road map

Let F be a finite field of order n. Then, $\text{char}(F) = p$ where p is prime. Then, $\mathbb{Z}_p$ is a subfield of F. F is a vector space over $\mathbb{Z}_p$ of dim $= k$. Then, order of F is p^k.

2020-01-17

DEFINITION 2.1.14: Isomorphic

We say two fields F and S are **isomorphic** if they have the same binary operations and if there exists a bijection between them.

DEFINITION 2.1.15: Subfield

Let F be a field. A subset $S \subseteq F$ is called a **subfield** of F if S is a field itself with respect to the same operations of F.

EXAMPLE 2.1.16: Subfield

Let F be a finite field where $\text{char}(F) = p$. Consider $E = \{0, 1, 1+1, \dots, \underbrace{1+\cdots+1}_{p-1}\} \subseteq F$. We see that E is a field with the same field operations as F. Also, E has order p. If we label the elements of E naturally such that $\underbrace{1+\cdots+1}_{p-1} \longleftrightarrow p-1$, then

$$E = \{0, 1, 1+1, \dots, \underbrace{1+\cdots+1}_{p-1}\} = \mathbb{Z}_p = \{0, 1, 2, \dots, p-1\} \subseteq F$$

So E is isomorphic to $\mathbb{Z}_p$.

THEOREM 2.1.17

If F is a finite field of characteristic p, then $\mathbb{Z}_p$ is a subfield of F.

EXERCISE 2.1.18

Prove Theorem 2.1.17.

DEFINITION 2.1.19

Let F be a finite field, and consider $\mathbb{Z}_p \subseteq F$.

- Each $v \in F$ is vector.
- Each $c \in \mathbb{Z}_p$ is a scalar.
- Addition in F is defined by vector addition.
- Multiplication in F by elements in $\mathbb{Z}_p$ is defined by scalar multiplication.

THEOREM 2.1.20

If F is a finite field of characteristic p, then F is a vector space over $\mathbb{Z}_p$.

EXERCISE 2.1.21

Prove Theorem 2.1.20.

THEOREM 2.1.22

If F is a finite field of characteristic p, then

$$\text{ord}(F) = p^k$$

for some $k \in \mathbb{Z}_{\geqslant 1}$.

Proof of Theorem 2.1.22

Let k be the dimension of the vector space F over $\mathbb{Z}_p$. Let $\{\alpha_1, \dots, \alpha_k\}$ be a basis for F. Then, every element in F can be written as

$$c_1\alpha_1 + \cdots + c_k\alpha_k$$

where $c_i \in \mathbb{Z}_p$. For each α_i, there are p possible choices for c_i, hence $\text{ord}(F) = p^k$.

EXAMPLE 2.1.23

There is no field of order 6.

Question: Is there a finite field of order 4, 8, 9?

2.2 Irreducible Polynomials

DEFINITION 2.2.1: Set of all polynomials in x over F

Let F be a field. The **set of all polynomials in x over F** (polynomials with coefficients from F) is denoted $F[x]$. Addition and multiplication are both done in the usual way, with coefficient arithmetic in F.

EXAMPLE 2.2.2: Polynomial Modular Arithmetic

In $\mathbb{Z}_{11}$, $(2+5x+6x^2)+(3+9x+5x^2)=5+3x$.

THEOREM 2.2.3

Let F be a field. $F[x]$ is a commutative ring.

DEFINITION 2.2.4

Let F be a field and let $f \in F[x]$ with $\deg(f) \geqslant 1$. If $g, h \in F[x]$ with $f \mid (g-h)$, then we write

$$g \equiv h \pmod{f}$$

or equivalently, we can write $g - h = \ell f$ for some $\ell \in F[x]$.

THEOREM 2.2.5

Congruence is an equivalence relation.

DEFINITION 2.2.6: Equivalence class containing $g \in F[x]$

For a given $f \in F[x]$, the **equivalence class containing $g \in F[x]$** is

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

DEFINITION 2.2.7

For $g, h \in F[x]$, we define addition and multiplication as follows:

- Addition: $[g]+[h]=[g+h]$
- Multiplication: $[g][h]=[gh]$

THEOREM 2.2.8

(1) The set of all equivalence classes, denoted $F[x]/(f)$ where $f \in F[x]$ and $\deg(f) \geqslant 1$ is a commutative ring.

(2) The polynomials in $F[x]$ of degree less than degree of f are a system of distinct representatives of equivalence classes in $F[x]/(f)$.

Proof of Theorem 2.2.8

Let $g \in F[x]$. By division algorithm for polynomials we can write $g = \ell f + r$ where $\deg(r) < \deg(f)$. So, $g - r = \ell f$. Hence, $g \equiv r \pmod{f}$. Thus, $[g] = [r]$ and we have $\deg(r) < \deg(f)$. Also, if $r_1, r_2 \in F[x]$

with $r_1 \neq r_2$, and $\deg(r_1), \deg(r_2) < \deg(f)$, then

$$f \nmid (r_1 - r_2) \iff r_1 \not\equiv r_2 \pmod{f}$$

Thus, $[r_1] \neq [r_2]$.

2020-01-20

DEFINITION 2.2.9: Irreducible

Let F be a field, and $f \in F[x]$ of degree $n \geqslant 1$. f is **irreducible** over F if f cannot be written as $f = gh$, where $g, h \in F[x]$ and $\deg(g), \deg(n) \geqslant 1$.

EXAMPLE 2.2.10: Irreducible

- $x^2 + 1$ is irreducible over $\mathbb{R}$.
- $x^2 + 1$ is reducible over $\mathbb{C}$ since $(x + i)(x - i) = x + 1$.
- $x^2 + 1$ is reducible over $\mathbb{Z}_2$ since $(x + 1)^2 = x + 1$.
- $x^2 + 1$ is irreducible over $\mathbb{Z}_3$.

THEOREM 2.2.11

Let F be a field and $f \in F[x]$ of degree $n \geqslant 1$. $F[x]/(f)$ is a field if and only if f is irreducible over F.

Proof of Theorem 2.2.11

Note that $F[x]/(f)$ is a commutative ring.
($\Longleftarrow$) Suppose $g \in F[x]/(f)$ where $g \neq 0$ and $\deg(g) < \deg(f)$. Then, $\gcd(g, f) = 1$ and so by EEA for polynomials, there exists $s, t \in F[x]$ such that

$$gs + ft = 1$$

Reducing both sides modulo f gives

$$gs \equiv 1 \pmod{f}$$

So, $g^{-1} = s$. Hence, $F[x]/(f)$ is a field.

EXERCISE 2.2.12

Prove the forward direction of Theorem 2.2.11.

We need an irreducible polynomial $f \in \mathbb{Z}_p[x]$ of degree n. Then, $\mathbb{Z}[x]/(f)$ is a finite field of order p^n.

THEOREM 2.2.13

For any prime p and $n \in \mathbb{Z}_{\geqslant 2}$, there exists an irreducible polynomial of degree n over $\mathbb{Z}_p$.

THEOREM 2.2.14

There exists a finite field of order q if and only if q is a prime power.

EXAMPLE 2.2.15

Construct a finite field of order $2^2 = 4$.

Solution. Take $f(x) = x^2 + x + 1 \in \mathbb{Z}_2[x]$ which is irreducible over $\mathbb{Z}_2[x]$. Thus, the field is

$$\mathbb{Z}_2[x]/(x^2 + x + 1) = \{0, 1, x, x + 1\}$$

Examples of operations:
- $x + (x + 1) = 1$
- $x(x + 1) = x^2 + x = 1$
- $x^{-1} = x + 1$
- $1^{-1} = 1$
- $x^{-1} = x + 1$
- $(x + 1)^{-1} = x$

EXAMPLE 2.2.16

Construct a field of order $2^3 = 8$.
Solution. We need an irreducible polynomial of degree 3 over $\mathbb{Z}_2$. Take $f_1(x) = x^3 + x + 1$ which is irreducible over $\mathbb{Z}_2$. Then a field of order 8 is

$$F_1 = \mathbb{Z}_2[x]/(x^3 + x + 1) = \{0, 1, x, x + 1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}$$

Examples of operations:
- $x^2 + (x^2 + x + 1) = x + 1$
- $x^2(x^2 + x + 1) = x^4 + x^3 + x^2 = 1$
- $(x^2)^{-1} = x^2 + x + 1$
- $x^{-1} = x^2 + 1$

EXAMPLE 2.2.17

Construct a field of order $2^3 = 8$.
Solution. Take $f_2(x) = x^3 + x^2 + 1$. Then a field of order 8 is

$$F_2 = \mathbb{Z}_2[x]/(x^3 + x^2 + 1) = \{0, 1, x, x + 1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}$$

Examples of operations:
- $x^{-1} = x^2 + x$

Note: F_1 and F_2 are two different fields of order $2^3 = 8$, but they are isomorphic. That is, there is a bijection $\alpha : F_1 \to F_2$ such that

$$\alpha(a + b) = \alpha(a) + \alpha(b)$$
$$\alpha(ab) = \alpha(a)\alpha(b)$$

for all $a, b \in F_1$.

THEOREM 2.2.18

Any two finite fields of order q are isomorphic.

EXERCISE 2.2.19

Prove Theorem 2.2.18.

DEFINITION 2.2.20: Galois field of order q

We will denote the **Galois field of order** q by $GF(q)$.

We saw one representation of $GF(2^2)$ and two different representations of $GF(2^3)$.

2020-01-22

EXAMPLE 2.2.21

Construct $GF(2^4 = 16)$.
Solution. Take $f(x) = x^4 + x + 1 \in \mathbb{Z}_2[x]$.
- f has no roots in $\mathbb{Z}_2$ and hence no linear factors.
- Long division shows that $x^2 + x + 1 \nmid x^4 + x + 1$, so f has no irreducible quadratic factors.
- f is irreducible over $\mathbb{Z}_2$.

Thus, $GF(16) = \mathbb{Z}_2[x]/(x^4 + x + 1)$.

2.3 Properties of Finite Fields

PROPOSITION 2.3.1: Coprimeness and Divisiblity (CAD)

† *For all integers a, b and c, if $c \mid ab$ and $\gcd(a, c) = 1$, then $c \mid b$.*

LEMMA 2.3.2

† *For each integer $k \in [1, p-1]$,*

$$p \mid \binom{p}{k}$$

Proof of Lemma 2.3.2

We know that $\binom{p}{k} \in \mathbb{Z}$ and

$$\binom{p}{k} = \frac{p!}{(p-k)k!}$$

Since $k \geqslant 1$, then

$$\binom{p}{k} = \frac{p(p-1)\cdots(p-k+1)}{k!}$$

Therefore, $k!\binom{p}{k} = p(p-1)\cdots(p-k+1)$.
We note that $p \mid p(p-1)\cdots(p-k+1)$ and therefore $p \mid k!\binom{p}{k}$. Since p is prime and $p > k$, then $\gcd(p, k!) = 1$. Therefore, by Proposition 2.3.1

$$p \mid \binom{p}{k}$$

THEOREM 2.3.3: Frosh's Dream

Let $\alpha, \beta \in GF(q)$ where $\text{char}(GF(q)) = p$.

$$(\alpha + \beta)^p = \alpha^p + \beta^p$$

Proof of Theorem 2.3.3

$$(\alpha + \beta)^p = \alpha^p + \sum_{k=1}^{p-1} \binom{p}{k} \alpha^k \beta^{p-k} + \beta^p$$

By Lemma 2.3.2,

$$p \mid \binom{p}{k} \implies p\lambda_k = \binom{p}{k}$$

where $\lambda_k \in \mathbb{Z}$ for each $k \in [1, p-1]$. Hence,

$$\begin{aligned}\sum_{k=1}^{p-1} \binom{p}{k} \alpha^k \beta^{p-k} &= \sum_{k=1}^{p-1} (p\lambda_k) \alpha^k \beta^{p-k} \\ &= \sum_{k=1}^{p-1} (\underbrace{1 + \cdots + 1}_{p}) \lambda_k \alpha^k \beta^{p-k} \\ &= 0\end{aligned}$$

Thus, $(\alpha + \beta)^p = \alpha^p + \beta^p$.

COROLLARY 2.3.4

Let $\alpha, \beta \in GF(q)$ where char$(GF(q)) = p$.

$$(\alpha + \beta)^{p^m} = \alpha^{p^m} + \beta^{p^m}$$

for all $m \in \mathbb{Z}_{\geqslant 1}$.

Proof of Corollary 2.3.4

† We prove this result by induction on m, where $P(m)$ is the statement

$$(\alpha + \beta)^{p^m} = \alpha^{p^m} + \beta^{p^m}$$

Base Case: The statement $P(1)$ is given by

$$(\alpha + \beta)^p = \alpha^p + \beta^p$$

which is clearly true by Theorem 2.3.3.
Inductive Hypothesis: Assume

$$(\alpha + \beta)^{p^k} = \alpha^{p^k} + \beta^{p^k}$$

for an arbitrary integer $k \geqslant 1$.
Inductive Conclusion: We wish to prove $P(k+1)$ which is the statement

$$(\alpha + \beta)^{p^{k+1}} = \alpha^{p^{k+1}} + \beta^{p^{k+1}}$$

Starting with the expression on the left-hand side of $P(k+1)$, we obtain

$$\begin{aligned}(\alpha + \beta)^{p^{k+1}} &= [(\alpha + \beta)^p]^{p^k} \\ &= (\alpha^p + \beta^p)^{p^k} && \text{by Theorem 2.3.3} \\ &= (\alpha^p)^{p^k} + (\beta^p)^{p^k} && \text{by IH} \\ &= \alpha^{p^{k+1}} + \beta^{p^{k+1}}\end{aligned}$$

The result is true for $m = k+1$, and hence holds for all $m \in \mathbb{Z}_{\geqslant 1}$ by the Principle of Mathematical Induction.

THEOREM 2.3.5

Let $\alpha \in GF(q)$. Then

$$\alpha^q = \alpha$$

Proof of Theorem 2.3.5

If $\alpha = 0$, then $\alpha^q = 0 = \alpha$.
If $\alpha \neq 0$, let

$$\{a_1, a_2, \dots, a_{q-1}\}$$

be the distinct non-zero elements in $GF(q)$. Consider

$$\{\alpha a_1, \alpha a_2 \dots, \alpha a_{q-1}\}$$

These are all distinct because otherwise for some $i \neq j$, $\alpha a_i = \alpha a_j \implies a_i = a_j$ which is a contradiction. Hence,

$$\{\alpha a_1, \dots, \alpha a_{q-1}\} = \{a_1, \dots, a_{q-1}\}$$

This implies

$$\begin{aligned}
&(\alpha a_1) \cdots (\alpha a_{q-1}) = a_1 \cdots a_{q-1} \\
&\implies \alpha^{q-1}(a_1 \cdots a_{q-1}) = a_1 \cdots a_{q-1} \\
&\implies \alpha^{q-1} = 1
\end{aligned}$$

since a_i is non-zero for each $i \in [1, q-1]$. Thus, since $\alpha \neq 0$ we have $\alpha^q = \alpha$.

DEFINITION 2.3.6

Let $GF(q)^* = GF(q)/\{0\}$.

DEFINITION 2.3.7: Order of $\alpha \in GF(q)^*$

The **order of $\boldsymbol{\alpha} \in \boldsymbol{GF(q)}^*$**, denoted $\text{ord}(\alpha)$, is the smallest positive integer t such that $\alpha^t = 1$.

EXAMPLE 2.3.8

How many elements of order 1 are there in $GF(q)$?
Solution. $\alpha = 1$

EXAMPLE 2.3.9

Find $\text{ord}(x)$ in $GF(16) = \mathbb{Z}_2/(x^4 + x + 1)$.
Solution.

- $x^1 = x$
- $x^2 = x^2$
- $x^3 = x^3$
- $x^4 = x + 1$
- $x^5 = x^2 + x$
- $x^6 = x^3 + x^2$
- $x^7 = x^3 + x + 1$
- $x^8 = x^2 + 1$
- $x^9 = x^3 + x$
- $x^{10} = x^2 + x + 1$
- $x^{11} = x^3 + x^2 + x$
- $x^{12} = x^3 + x^2 + x + 1$
- $x^{13} = x^3 + x^2 + 1$
- $x^{14} = x^3 + 1$
- $x^{15} \equiv 1 \pmod{x^4 + x + 1}$

Since $\text{ord}(x) \neq 1, 3, 5$ $\text{ord}(x) \mid 15$, so we have $\text{ord}(x) = 15$.

LEMMA 2.3.10

Let $\alpha \in GF(q)^*$, $\text{ord}(\alpha) = t$ and $s \in \mathbb{Z}$.

$$\alpha^s = 1 \iff t \mid s$$

Proof of Lemma 2.3.10

Let $s \in \mathbb{Z}$. By the division algorithm for integers,

$$s = \ell t + r$$

where $0 \leqslant r \leqslant t-1$. Then

$$\alpha^s = \alpha^{\ell t + r} = (\alpha^t)^\ell \alpha^r = \alpha^r$$

So,

$$\begin{aligned} \alpha^s = 1 &\iff a^r = 1 \\ &\iff r = 0 \qquad \text{since } 0 \leqslant r \leqslant t-1 \\ &\iff t \mid s \end{aligned}$$

COROLLARY 2.3.11

If $\alpha \in GF(q)^*$, then $\text{ord}(\alpha) \mid (q-1)$.

Proof of Corollary 2.3.11

We know $\alpha^{q-1} = 1$, so $\text{ord}(\alpha) \mid (q-1)$ by the previous Lemma.

DEFINITION 2.3.12: Generator

An element $\alpha \in GF(q)$ is a **generator** of $GF(q)^*$ if

$$\{\alpha^i : i \geqslant 0\} = GF(q)^*$$

That is, α generates all the non-zero field elements. $\text{ord}(\alpha) = q-1$.

THEOREM 2.3.13

If α is a generator of $GF(q)^*$, then

$$\{\alpha^1, \ldots, \alpha^{q-1}\} = GF(q)^*$$

2020-01-24

THEOREM 2.3.14

If $GF(q)^*$ has order t, then

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

are pairwise distinct.

Proof of Theorem 2.3.14

Suppose for a contradiction that $\alpha^i = \alpha^j$ where $0 \leqslant i, j \leqslant t-1$. WLOG suppose $j > i$, then $\alpha^{j-i} = 1$ which contradicts $\text{ord}(\alpha) = t$ since $1 \leqslant j - i \leqslant t-1$.

2.4 † Existence of Generators

LEMMA 2.4.1

Let $\alpha \in GF(q)^*$ with $\text{ord}(\alpha) = t$. Then $\text{ord}(\alpha^i) = t/\gcd(t, i)$.

Proof of Lemma 2.4.1

Let $d = \gcd(t, i)$. The order of a^i is the smallest positive integer s such that $\alpha^{is} = 1$. Now,

$$\alpha^{is} = 1 \iff t \mid is \iff \frac{t}{d} \mid \frac{i}{d}s \iff \frac{t}{d} \mid s$$

Since the smallest positive integer s satisfying $\frac{t}{d} \mid s$ is $s = \frac{t}{d}$, we have $\text{ord}(\alpha^i) = \frac{t}{d}$.

LEMMA 2.4.2

Let $\alpha, \beta \in GF(q)^*$, with $\text{ord}(\alpha) = m$ and $\text{ord}(\beta) = n$. If $\gcd(m, n) = 1$ then $\text{ord}(\alpha\beta) = mn$.

Proof of Lemma 2.4.2

Let $t = \text{ord}(\alpha\beta)$. Now,

$$(\alpha\beta)^{mn} = \alpha^{mn}\beta^{mn} = 1,$$

so $t \mid mn$. Also,

$$1 = (\alpha\beta)^{tn} = \alpha^{tn}\beta^{tn} = \alpha^{tn},$$

so $m \mid tn$. And, since $\gcd(m, n) = 1$, we have $m \mid t$. Similarly,

$$1 = (\alpha\beta)^{tm} = \alpha^{tm}\beta^{tm} = \beta^{tm},$$

so $n \mid tm$. And, since $\gcd(m, n) = 1$, we have $n \mid t$. Hence, since $\gcd(m, n) = 1$, we have $mn \mid t$. Thus. $t = mn$.

THEOREM 2.4.3

Every finite field $GF(q)$ has a generator.

Proof of Theorem 2.4.3

Let α be an element of highest order in $GF(q)^*$; say $\text{ord}(\alpha) = t$. Suppose that $t < (q - 1)$.
If the order of every element in $GF(q)^*$ were to divide t then the equation $y^t - 1 = 0$ would have $q - 1$ roots in $GF(q)$, which is impossible since $(q - 1) > t$. Hence, there exists an element $\beta \in GF(q)^*$ whose order b does not divide t.
Now, let ℓ be a prime such that the highest power of ℓ which divides b (say ℓ^e) is greater than the highest power of ℓ which divides t (say ℓ^f)—such a prime ℓ must exist since b does not divide t.
Consider the field elements $\alpha' = \alpha^{\ell^f}$ and $\beta' = \beta^{b/\ell^e}$. We have

$$\text{ord}(\alpha') = \frac{t}{\gcd(t, \ell^f)} = \frac{t}{\ell^f}$$

and

$$\text{ord}(\beta') = \frac{b}{\gcd(b, \ell^e)} = \frac{b}{b/\ell^e} = \ell^e$$

Since $\gcd(t/\ell^f, \ell^e) = 1$, we have $\text{ord}(\alpha'\beta') = (t/\ell^f)(\ell^e) = t\ell^{e-f} > t$. This contradicts the hypothesis that the highest order of any element in $GF(q)^*$ is t. Hence, the hypothesis that $t < (q - 1)$ is wrong, and so $t = q - 1$. Thus, α is a generator of $GF(q)^*$.

Chapter 3

Linear Codes

3.1 Introduction

Let $F = GF(q)$. Let $V_n(F) = \underbrace{F \times \cdots \times F}_{n} = F^n$. Then, $V_n(F)$ is an n-dimensional vector space over F, and we have $|V_n(F)| = q^n$.

DEFINITION 3.1.1: Linear (n, k)-code

Let $F = GF(q)$. A **linear (n, k)-code** over F is a k-dimensional subspace of $V_n(F)$.

DEFINITION 3.1.2: Subspace

A **subspace** of a vector space V over F is a subset $S \subseteq V$ such that
V1 $\mathbf{0} \in S \implies S \neq \emptyset$
V2 $\boldsymbol{v}_1 + \boldsymbol{v}_2 \in S$, $\forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in S$
V3 $\lambda \boldsymbol{v} \in S$, $\forall \lambda \in F$ and $\boldsymbol{v} \in S$
Note that $S \subseteq V$ is also a vector space over F.

Let C be an (n, k)-code over F. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ be an ordered basis for C.

(1) The codewords in C are precisely:
$$m_1 \boldsymbol{v}_1 + \cdots + m_k \boldsymbol{v}_k$$
where $m_i \in F$. So, $|C| = M = q^k$ since there are q choices for each m. The length of C is n and has dimension k.

(2) The rate of C is
$$R = \frac{\log_q(M)}{n} = \frac{k}{n}$$

DEFINITION 3.1.3: Hamming weight

The **Hamming weight** of $\boldsymbol{v} \in V_n(F)$, denoted $w(\boldsymbol{v})$ is the number of non-zero coordinate positions in $\boldsymbol{v}$.

DEFINITION 3.1.4: Hamming weight of an (n, k)-code

The **Hamming weight of an (n, k)-code** C is:
$$w(C) = \min\{w(\boldsymbol{c}) : \boldsymbol{c} \in C, \boldsymbol{c} \neq \mathbf{0}\}$$

THEOREM 3.1.5

If C is a linear code, then $d(C) = w(C)$.

Proof of Theorem 3.1.5

$$\begin{aligned} d(C) &= \min\{d(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x}, \boldsymbol{y} \in C,\ \boldsymbol{x} \neq \boldsymbol{y}\} \\ &= \min\{w(\boldsymbol{x} - \boldsymbol{y}) : \boldsymbol{x}, \boldsymbol{y} \in C,\ \boldsymbol{x} \neq \boldsymbol{y}\} && \text{by A2Q1a} \\ &= \min\{w(\boldsymbol{c}) : \boldsymbol{c} \in C,\ \boldsymbol{c} \neq \boldsymbol{0}\} && \text{since } C \text{ is a vector space} \\ &= w(C) \end{aligned}$$

3.2 Generator Matrices and the Dual Code

Since $M = q^k$, there are q^k source messages. We'll assume that the source messages are elements of $V_k(F)$. Then, a natural encoding rule is, given $\begin{bmatrix} m_1 & \dots & m_k \end{bmatrix}_{1\times k} \in V_k(F)$ we'll encode the message as

$$\boldsymbol{c} = m_1\boldsymbol{v}_1 + \cdots + m_k\boldsymbol{v}_k$$

The encoding rule depends on the basis chosen for C.

If $\boldsymbol{m} = \begin{bmatrix} m_1 & \dots & m_k \end{bmatrix}_{1\times k}$, then the encoding rule can be written as follows:

$$\begin{aligned} \boldsymbol{c} &= \begin{bmatrix} m_1 & \dots & m_k \end{bmatrix}_{1\times k} \begin{bmatrix} -\boldsymbol{v}_1- \\ -\boldsymbol{v}_2- \\ \vdots \\ -\boldsymbol{v}_k- \end{bmatrix}_{k\times n} \\ &= \boldsymbol{m}G \end{aligned}$$

Note that $\boldsymbol{v}_i$ are row vectors in this course; that is,

$$\boldsymbol{v}_i = \begin{bmatrix} v_{i1} & \dots & v_{in} \end{bmatrix}_{1\times n}$$

DEFINITION 3.2.1: Generator matrix

Let C be an (n, k)-code. A **generator matrix** G for C is a $k \times n$ matrix whose rows form a basis for C.

Note: An encoding rule for C with respect to G is $\boldsymbol{c} = \boldsymbol{m}G$. Performing elementary row operations on G gives a different matrix for the same code C due to the order of the basis.

2020-01-27

EXAMPLE 3.2.2

Consider a $\underbrace{\text{binary}}_{F=GF(2)=\mathbb{Z}_2}$ $(\underset{n}{5}, \underset{k}{3})$-code C where $C = \{\underbrace{10010}_{\boldsymbol{v}_1}, \underbrace{01011}_{\boldsymbol{v}_2}, \underbrace{00101}_{\boldsymbol{v}_3}\}$.
A possible generator matrix of a C is the following.

$$G = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right]_{3\times 5}$$

$\text{rank}(G) = 3$, thus G is a generator matrix for C.

Using the encoding rule, we can determine some possible codewords in C.

Message ($\boldsymbol{m}$)	$\rightarrow$	Codeword ($\boldsymbol{c}$)
000	$\rightarrow$	00000
001	$\rightarrow$	00101
010	$\rightarrow$	01011
011	$\rightarrow$	01110
100	$\rightarrow$	10010
101	$\rightarrow$	10111
110	$\rightarrow$	11001
111	$\rightarrow$	11100

- $M = q^k = 2^3$
- $R = {}^k/_n = {}^3/_5$
- $d(C) = 2$
- $e = 0$

Note: Any matrix equivalent to G is also a generator matrix for C, but yields a different encoding rule.

DEFINITION 3.2.3: Systematic, Standard form

Let $\left[\, I_k \mid A \,\right]_{k \times n}$ be a generator matrix for an (n, k)-code C. If an (n, k)-code has a generator matrix of this form, then C is **systematic** and the generator matrix is in **standard form**.

EXAMPLE 3.2.4

$C = \{100011, 101010, 100110\}$ is a non-systematic $(6, 3)$-code.
Some generator matrices are:

$$G_1 = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right]$$

$G_1 : R_2 + R_1$

$$G_2 = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right]$$

$G_2 : R_3 + R_1$

$$G_3 = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right]$$

Clearly C is not systematic. However, if every codeword is permuted by moving the second bit to the fourth bit, we get C' that is linear and has the same length, dimension, and distance as C.

DEFINITION 3.2.5: Equivalent code

Let C be an (n, k)-code. If π is a permutation on $\{1, \ldots, n\}$. Then $\pi(C)$ (that is, apply π to each codeword) is an (n, k)-code which is said to be an **equivalent code** for C.

THEOREM 3.2.6

(1) If C and C' are equivalent codes, then

$$d(C) = d(C')$$

(2) Every linear code is equivalent to a systematic code.

Proof of Theorem 3.2.6

Let C be an (n, k) code. Let G be a generator matrix for C in RREF. Then, one can permute the columns of G to get a matrix $G' = \left[\, I_k \mid A \,\right]$ in standard form. Then, G' is a generator matrix for a code C' that is equivalent to C.

DEFINITION 3.2.7: Inner product

Let $\boldsymbol{x}, \boldsymbol{y} \in V_n(F)$. The **inner product** of $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i \in F$$

THEOREM 3.2.8

If $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in V_n(F)$ and $\lambda \in F$, then
(1) $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{y} \cdot \boldsymbol{x}$
(2) $\boldsymbol{x} \cdot (\boldsymbol{y} + \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{y} + \boldsymbol{x} \cdot \boldsymbol{z}$
(3) $(\lambda \boldsymbol{x}) \cdot \boldsymbol{y} = \lambda(\boldsymbol{x} \cdot \boldsymbol{y})$
(4) $\boldsymbol{x} \cdot \boldsymbol{x} = \boldsymbol{0}$ *does* ***not*** *imply* $\boldsymbol{x} = \boldsymbol{0}$

EXAMPLE 3.2.9

Consider $V_2(\mathbb{Z}_2)$. Then, $\begin{bmatrix} 1 & 1 \end{bmatrix}_{1\times 2} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}_{1\times 2} = 0$.

DEFINITION 3.2.10: Dual code

Let C be an (n, k)-code over F. The **dual code** of C is

$$C^{\perp} = \{\boldsymbol{x} \in V_n(F) : \boldsymbol{x} \cdot \boldsymbol{c} = \boldsymbol{0} \quad \forall \boldsymbol{c} \in C\}$$

THEOREM 3.2.11

Let $\boldsymbol{x} \in V_n(F)$.

$$\boldsymbol{x} \in C^{\perp} \iff \boldsymbol{v}_1 \cdot \boldsymbol{x} = \cdots = \boldsymbol{v}_k \cdot \boldsymbol{x} = 0$$

Proof of Theorem 3.2.11

($\Longrightarrow$) If $\boldsymbol{x} \in C^{\perp}$, then $\boldsymbol{x} \cdot \boldsymbol{c} = 0$ for all $\boldsymbol{c} \in C$. In particular,

$$\boldsymbol{x} \cdot \boldsymbol{v}_1 = \cdots = \boldsymbol{x} \cdot \boldsymbol{v}_k = 0$$

($\Longleftarrow$) Suppose $\boldsymbol{x} \cdot \boldsymbol{v}_1 = \cdots = \boldsymbol{x} \cdot \boldsymbol{v}_k = 0$. Let $\boldsymbol{c} \in C$. We can write

$$\boldsymbol{c} = \lambda_1 \boldsymbol{v}_1 + \cdots + \lambda_k \boldsymbol{v}_k$$

for all $\lambda_i \in F$. Then,

$$\boldsymbol{x} \cdot \boldsymbol{c} = \lambda_1(\boldsymbol{x} \cdot \boldsymbol{v}_1) + \cdots + \lambda_k(\boldsymbol{x} \cdot \boldsymbol{v}_k) = 0$$

Hence, $\boldsymbol{x} \in C^{\perp}$.

THEOREM 3.2.12

If C is an (n,k)-code over F, then $C^{\perp}$ is an $(n, n-k)$-code over F.

Proof of Theorem 3.2.12

Consider

$$G = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_k \end{bmatrix}_{k \times n}$$

Then, $\boldsymbol{x} \in C^{\perp}$ if and only if $G\boldsymbol{x}^{\top} = \boldsymbol{0}$. So, $C^{\perp}$ is the null space of G. Hence, $C^{\perp}$ is an $(n-k)$-dimensional subspace of $V_n(F)$.

2020-01-29

DEFINITION 3.2.13: Orthogonal

If $\boldsymbol{x}, \boldsymbol{y} \in V_n(F)$ and $\boldsymbol{x} \cdot \boldsymbol{y} = 0$, then $\boldsymbol{x}$ and $\boldsymbol{y}$ are **orthogonal**.

THEOREM 3.2.14

If C is a linear code, then $(C^{\perp})^{\perp} = C$.

Proof of Theorem 3.2.14

Let C be an (n,k)-code. Then $C^{\perp}$ is an $(n, n-k)$-code. So, $(C^{\perp})^{\perp}$ is an (n,k)-code. But $C \subseteq (C^{\perp})^{\perp}$ by definition of $C^{\perp}$. Suppose C is a code over $F = GF(q)$. Then $|C| = q^k$ and $|(C^{\perp})^{\perp}| = q^k$. Thus, $C = (C^{\perp})^{\perp}$.

THEOREM 3.2.15

Let C be an (n,k)-code with standard form $k \times n$ generator matrix. Then, a generator matrix for $C^{\perp}$ is

$$H = \left[-A^{\top} \mid I_{n-k} \right]_{(n-k) \times n}$$

Proof of Theorem 3.2.15

$\text{rank}(H) = n - k$, so H is indeed a generator matrix for some $(n, n-k)$-code $\overline{C}$. Now,

$$\begin{aligned} GH^{\top} &= \left[I_k \mid A \right]_{k \times n} \begin{bmatrix} -A \\ I_{n-k} \end{bmatrix}_{n \times (n-k)} \\ &= -A + A \\ &= 0 \end{aligned}$$

Since $GH^{\top} = 0$, every row of H is orthogonal to every row of G, so every vector in the row space of H is orthogonal to every vector in the row space of G. Hence, $\overline{C} \subseteq C^{\top}$. Since $\dim(\overline{C}) = \dim(C^{\perp})$ we have $\overline{C} = C^{\perp}$.

3.3 The Parity-Check Matrix

DEFINITION 3.3.1: Parity-check matrix

A generator matrix for $C^\perp$ is called a **parity-check matrix** (PCM) for C.

EXAMPLE 3.3.2

Consider a $(5,2)$-code C over $\mathbb{Z}_3$ with generator matrix

$$G = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \leftarrow \boldsymbol{c}_1 \\ \leftarrow \boldsymbol{c}_2 \end{matrix}$$

Find the length, dimension, order, number of codewords, codewords, distance, weight and errors that can be corrected for C.

Solution.

- Length: $n = 5$ ((n,k)-code)
- Dimension: $k = 2$ ((n,k)-code)
- Order: $q = 3$ ($\mathbb{Z}_3$)
- Number of codewords: $M = q^k = 3^2 = 9$
- Codewords: $C = \{00000, 20210, 10120, 11001, 22002, 01211, 12212, 21121, 02122\}$
- Distance: $d(C) = w(C) = 3$
- Error-correcting capability: $e = 1$

Find a generator matrix for $C^\perp$.

Solution.

$$\begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 1 & 1 \end{bmatrix}$$

So,

$$H = \left[\begin{array}{cc|ccc} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{array}\right]$$

is a generator matrix for $C^\perp$ which is a $(5,3)$-code over $\mathbb{Z}_3$. $M = 3^3 = 27$.

2020-01-31

THEOREM 3.3.3

Let C be an (n,k)-code over F, and let H be a PCM for C. Then $d(C) \geqslant s$ if and only if every $(s-1)$ columns of H are linearly independent over F.

Proof of Theorem 3.3.3

Let $h_1, \dots, h_n$ be the columns of H.

($\impliedby$) Suppose $d(C) \leqslant s-1$. By Theorem 3.1.5, we have that $w(C) \leqslant s-1$. Let $\boldsymbol{c} \in C$, with $1 \leqslant w(c) \leqslant s-1$. WLOG, suppose $c_j = 0$ for each $j \in [s,n]$. Since $\boldsymbol{c} \in C$, we have $H\boldsymbol{c}^\top = 0$. Therefore, $c_1 h_1 + \cdots + c_{s-1} h_{s-1} = 0$. Since $w(C) \geqslant 1$, this is a non-trivial linear combination of $h_1, \dots, h_{s-1}$ that equal 0. So, $h_1, \dots, h_{s-1}$ are linearly dependent over F.

($\implies$) Suppose there are $s-1$ columns of H that are linearly dependent over F, say $h_1, \dots, h_{s-1}$. So, we can write

$$c_1 h_1 + \cdots + c_{s-1} h_{s-1}$$

where $c_j \in F$ not all zero for each $j \in [1, s-1]$. Let $\boldsymbol{c} = (c_1, \dots, c_{s-1}, \underbrace{0, \dots, 0}_{n-s+1}) \in V_n(F)$. Then, $H\boldsymbol{c}^\top = 0$.

So, $\boldsymbol{c} \in C$ where $1 \leqslant w(\boldsymbol{c}) \leqslant s-1$. Thus, $d(C) \leqslant s-1$.

COROLLARY 3.3.4

Let C be an (n,k)-code over F with PCM H. Then, $d(C)$ is the smallest number of columns of H that are linearly dependent over F.

EXAMPLE 3.3.5

Recall, we found a PCM

$$H = \left[\begin{array}{cc|ccc} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{array}\right]$$

for a $(5,2)$-code C over $\mathbb{Z}_3$. Find $d(C)$.
Solution.

- No 0 column in $H \implies d(C) \geqslant 2$.
- No two linearly dependent columns in H since there are no repeated columns, and no column is a scalar multiple of another column $\implies d(C) \geqslant 3$.
- Three columns are linearly dependent as seen in the following equation.

$$\begin{bmatrix} 2 & 1 & 0 \end{bmatrix}^\top = 2\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top$$

Thus, $d(C) = 3$.

EXAMPLE 3.3.6

Let C be a binary code with PCM H.

- $d(C) = 1 \iff H$ has a 0 column.
- $d(C) = 2 \iff$ the columns of H are non-zero and two are the same.
- $d(C) = 3 \iff$ the columns of H are non-zero, distinct, and one column is the sum of two other (distinct) columns.

3.4 Hamming Codes and Perfect Codes

EXAMPLE 3.4.1

Construct a $(7,4,3)$-binary code C.
Solution. Consider a PCM for C:

$$H = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]_{3\times 7}$$

This is a **Hamming Code** of order 3 over $GF(2)$.

DEFINITION 3.4.2: Hamming bound

Let C be an $[n,M]$-code with distance d over an alphabet A of size q. Let $e = \left\lfloor \frac{d-1}{2} \right\rfloor$. The **sphere packing bound** or **Hamming bound** is:

$$M \sum_{i=0}^{e} \binom{n}{i} (q-1)^i \leqslant q^n$$

DEFINITION 3.4.3: Perfect Code

Let C be an $[n, M]$-code over A of distance d. Then, C is a **perfect code** if

$$M \sum_{i=0}^{e} \binom{n}{i} (q-1)^i = q^n$$

Note: If C is perfect, then IMLD=CMLD.

2020-02-03

For fixed n, q, d, a perfect code maximizes

$$R = \frac{\log_q(M)}{n}$$

EXAMPLE 3.4.4

- $GF(q)^n$ is a trivial perfect code with $d = 1$.
- $C = \{\underbrace{0 \cdots 0}_{n}, \underbrace{1 \cdots 1}_{n}\}$ over $\mathbb{Z}_2$ is a perfect code if and only if n is odd.

EXERCISE 3.4.5

Prove that every perfect code must have odd distance (without referring to Theorem 3.4.6).

Proof of Exercise 3.4.5

Let C be an even code of even distance $d = 2t$. Then, $e = \lfloor (d-1)/2 \rfloor = t - 1$. Let $\boldsymbol{c} \in C$ and $\boldsymbol{r}$ be a vector such that $d(\boldsymbol{c}, \boldsymbol{r}) = t$. Note that $\boldsymbol{r}$ is not in the sphere of radius e centred at $\boldsymbol{c}$. Now, if $\boldsymbol{r}$ were in the sphere of radius e centred at some codeword $\boldsymbol{c}' \neq \boldsymbol{c}$, then we would have

$$d(\boldsymbol{c}, \boldsymbol{c}') \leqslant d(\boldsymbol{c}, \boldsymbol{r}) + d(\boldsymbol{r}, \boldsymbol{c}') \leqslant t + e < d$$

which is impossible since the distance of C is d. Hence, $\boldsymbol{r}$ is not contained in any of the radius-e spheres centred at codewords, and so C is not a perfect code. It follows that a perfect code must have odd distance.

THEOREM 3.4.6: Tietäväinen, 1973

The only perfect codes are:

(1) $V_n(GF(q))$.

(2) The binary replication code of odd length.

(3) The $(23, 12, 7)$*-binary Golay code and all codes equivalent to it.*

(4) The $(11, 6, 5)$*-ternary Golay code and all codes equivalent to it. A generator matrix for this code is:*

$$G = \left[\begin{array}{c|ccccc} & 1 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 2 & 2 & 1 \\ & 1 & 0 & 1 & 2 & 2 \\ I_6 & 2 & 1 & 0 & 1 & 2 \\ & 2 & 2 & 1 & 0 & 1 \\ & 1 & 2 & 2 & 1 & 0 \end{array}\right]_{6 \times 11}$$

(5) The Hamming codes and all codes of the same $[n, M, d]$ *parameters as them with* $d = 3$.

DEFINITION 3.4.7: Hamming code of order r over $GF(q)$

A **Hamming code of order r over $GF(q)$** is a linear code over $GF(q)$ with $n = \dfrac{q^r - 1}{q - 1}$, $k = n - r$ and an $r \times n$ PCM matrix whose columns are non-zero, and no two columns are scalar multiples of each other.

EXAMPLE 3.4.8

A Hamming code of order $r = 3$ over $GF(3)$ is a $(13, 10, 3)$-code over $GF(3)$ with PCM:

$$H = \left[\begin{array}{c|cccccccccc} & 1 & 0 & 1 & 2 & 2 & 0 & 2 & 1 & 2 & 1 \\ I_3 & 1 & 1 & 0 & 1 & 0 & 1 & 2 & 2 & 1 & 1 \\ & 0 & 1 & 1 & 0 & 1 & 2 & 1 & 2 & 2 & 1 \end{array}\right]_{3\times 13}$$

Observations:

(i) For every non-zero vector $\boldsymbol{v} \in V_r(GF(q))$, exactly one scalar multiple of $\boldsymbol{v}$ must be a column of a PCM (for the Hamming code of order r over $GF(q)$).

(ii) The dimension of the code is indeed k since rank(PCM) $= r = n - k$ since $\lambda_i e_i$ are columns of the PCM.

(iii) The Hamming codes have distance 3.

THEOREM 3.4.9

Hamming codes are perfect.

Proof of Theorem 3.4.9

Recall that Hamming codes have $e = 1$ and $n = \frac{q^r - 1}{q - 1}$ with $r = n - k$.

$$\begin{aligned} M \sum_{i=0}^{e} \binom{n}{i} (q-1)^i &= q^{n-r}(1 + n(q-1)) \\ &= q^{n-r}\left(1 + \frac{q^r - 1}{q-1}(q-1)\right) \\ &= q^{n-r}(1 + q^r - 1) \\ &= q^{n-r} q^r \\ &= q^n \end{aligned}$$

DEFINITION 3.4.10: Error vector

Suppose $\boldsymbol{c} \in C$ is transmitted. Suppose $\boldsymbol{r} \in V_n(F)$ is received. Then, the **error vector** is $\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{c}$.

EXAMPLE 3.4.11: Error Vector

Over $\mathbb{Z}_3$, if $\boldsymbol{c} = (120212)$ is sent, and $\boldsymbol{r} = (122102)$ is received, then the error vector is $\boldsymbol{e} = (002220)$.

3.5 Decoding Single-Error Correcting Codes

Let H be a PCM for an (n,k)-code C over $GF(q)$ with $d \geqslant 3$.

$$\begin{aligned} H\boldsymbol{r}^\top &= H(\boldsymbol{c}+\boldsymbol{e})^\top \\ &= H\boldsymbol{c}^\top + H\boldsymbol{e}^\top \\ &= H\boldsymbol{e}^\top && \text{since } \boldsymbol{c}^\top \text{ is in null space of } H \end{aligned}$$

DEFINITION 3.5.1: Syndrome

Let H be a parity-check matrix for an (n,k)-code. The **syndrome** $\boldsymbol{s}$ of $\boldsymbol{r}$ is defined to be $\boldsymbol{s} = H\boldsymbol{r}^\top$.

Notes:

(1) $\boldsymbol{r}$ and $\boldsymbol{e}$ have the same syndrome.

(2) If $\boldsymbol{e} = \boldsymbol{0}$, then $H\boldsymbol{e}^\top = \boldsymbol{0}$.

(3) If $w(\boldsymbol{e}) = 1$, say $\boldsymbol{e} = (0,\dots,0,\alpha,0,\dots,0)$ where α is in the i^{th} position with $\alpha \neq 0$, then $H\boldsymbol{e}^\top = \alpha h_i$ where h_i is the i^{th} column of H.

(4) The converse of (2) and (3) are false.

Algorithm 1: Decoding Algorithm for Single-Error Correcting Codes

Input : Parity-check matrix $H = (\boldsymbol{h}_1,\dots,\boldsymbol{h}_n)^\top$ and received vector $\boldsymbol{r}$
Output : Decoded vector

```
s ← Hr^⊤
if w(s) = 0 then
    return r
for i ← 0 to n do
    if s = αh_i with α ≠ 0 then
        return r − e
return
```

REMARK 3.5.2

$\boldsymbol{h}_i$ are column vectors in the input for each $i \in [1,n]$.

Claim: If $w(\boldsymbol{e}) \leqslant 1$, then the decoding algorithm always makes the correct decision.

Note: If H is a Hamming code and $w(\boldsymbol{e}) \geqslant 2$, then this decoding algorithm will always make the wrong decision.

EXAMPLE 3.5.3: Single-Error Decoding

Consider the $(7,4,3)$-binary Hamming code with PCM

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}_{3\times 7}$$

Decode $\boldsymbol{r} = (0111110)$.

Solution.

1. $\boldsymbol{s} = H\boldsymbol{r}^\top = (011)$.
2. $\boldsymbol{s}$ is the 6th column of H, so $\boldsymbol{e} = (0000010)$.

3. $\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = (0111100)$.

Verify that $H\boldsymbol{c}^\top = \mathbf{0}$.

General Decoding Problem for Binary Linear Codes

Instance: An $(n-k) \times n$ matrix H over $GF(2)$ with rank$(H) = n-k$. $\boldsymbol{r} \in V_n(GF(2))$.

Find: A vector $\boldsymbol{e} \in V_n(GF(2))$ of minimum weight with $H\boldsymbol{r}^\top = H\boldsymbol{e}^\top$.

Fact: This problem is NP-hard.

- P = problems solvable in polynomial time; that is, efficiently.
- NP = a certain class of problems including problems of strong practical interest which we do not know how to solve efficiently.
- NP-hard = If any single problem in this class of problems can be solved efficiently, then so can all problems in NP, in which case P=NP.

2020-02-07

3.6 Decoding Linear Codes

DEFINITION 3.6.1

Let C be an (n,k)-code over $F = GF(q)$ with PCM H. We write $\boldsymbol{x} \equiv \boldsymbol{y} \pmod{C}$, where $\boldsymbol{x}, \boldsymbol{y} \in V_n(F)$ if $(\boldsymbol{x} - \boldsymbol{y}) \in C$.

Notes:

(1) Congruence is an equivalence relation. That is, it has the following three properties:

(i) Reflexivity

(ii) Symmetry

(iii) Transitivity

(2) The set of equivalence classes partitions $V_n(F)$.

(3) The equivalence classes containing $\boldsymbol{x} \in V_n(F)$ is called a **coset** of $V_n(F)$. This class is:

$$\begin{aligned}\{\boldsymbol{y} \in V_n(F) : \boldsymbol{y} \equiv \boldsymbol{x} \pmod{C}\} &= \{\boldsymbol{x} + \boldsymbol{c} : \boldsymbol{c} \in C\} \\ &= C + \boldsymbol{x}\end{aligned}$$

We call $C + \boldsymbol{x}$ the coset of C represented by $\boldsymbol{x}$.

EXAMPLE 3.6.2: Cosets

Consider a $(5,2)$-binary code with generator matrix

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}_{2\times 5}$$

with $d(C) = 3$. Find all cosets of C.

Solution. The cosets of C are:

(1) $C + 00000 = \{00000, 10111, 01110, 11001\} = \{\mathbf{0}, R_1, R_2, R_1 + R_2\} = C + 10111 = C + 01110 = C + 11001$

(2) $C + 10000 = \{10000, 00111, 11110, 01001\}$

(3) $C + 01000 = \{01000, 11111, 00000, 10001\}$

(4) $C + 00100 = \{00100, 10011, 01010, 11101\}$
(5) $C + 00010 = \{00010, 10101, 01100, 11011\}$
(6) $C + 00001 = \{00001, 10110, 01111, 11000\}$
(7) $C + 00011 = \{00011, 10100, 01101, 11010\}$
(8) $C + 11100 = \{11100, 01011, 10010, 00101\}$
In total, there are 8 cosets.

Notes:

(1) $C + \mathbf{0} = C$.

(2) If $\boldsymbol{y} \in C + \boldsymbol{x}$, then $C + \boldsymbol{y} = C + \boldsymbol{x}$ by definition of equivalence relation—more specifically symmetry.

(3) The number of cosets is $q^n/q^k = q^{n-k}$.

Recall: If $\boldsymbol{x} \in V_n(F)$, then it's syndrome is

$$\boldsymbol{s} = H\boldsymbol{r}^\top \in V_{n-k}(F)$$

THEOREM 3.6.3

Let $\boldsymbol{x}, \boldsymbol{y} \in V_n(F)$. Then $\boldsymbol{x} \equiv \boldsymbol{y} \pmod{C}$ if and only if $H\boldsymbol{x}^\top = H\boldsymbol{y}^\top$.

Proof of Theorem 3.6.3

$$\begin{aligned} \boldsymbol{x} \equiv \boldsymbol{y} \pmod{C} &\iff (\boldsymbol{x} - \boldsymbol{y}) \in C \\ &\iff H(\boldsymbol{x} - \boldsymbol{y})^\top = \mathbf{0} \\ &\iff H\boldsymbol{x}^\top = H\boldsymbol{y}^\top \end{aligned}$$

So, cosets are characterized by their syndromes.

Decoding

- $\boldsymbol{c} \in C$ is sent.
- $\boldsymbol{r} \in V_n(F)$ is received.
- $\boldsymbol{e} = (\boldsymbol{r} - \boldsymbol{c}) \in V_n(F)$.
- $H\boldsymbol{r}^\top = H\boldsymbol{e}^\top$.

So, $\boldsymbol{r}$ and $\boldsymbol{e}$ belong to the same coset of C.

CMLD

Given $\boldsymbol{r}$, find a vector $\boldsymbol{e}$ of smallest weight in $C + \boldsymbol{r}$ or equivalently, find a vector $\boldsymbol{e}$ of smallest weight with the same syndrome as $\boldsymbol{r}$. Then, decode $\boldsymbol{r}$ to $\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e}$.

IMLD

Find the unique vector $\boldsymbol{e}$ of smallest weight in $C + \boldsymbol{r}$ having the same syndrome as $\boldsymbol{r}$. If no such $\boldsymbol{e}$ exists, then reject $\boldsymbol{r}$. Otherwise, decode $\boldsymbol{r}$ to $\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e}$.

3.7 Syndrome Decoding Algorithm

Given a PCM H for an (n,k)-code C over $F = GF(q)$.

DEFINITION 3.7.1: Coset leader

A vector of smallest weight is a coset of C is distinguished and called a **coset leader** (of that coset).

Algorithm 2: Syndrome Decoding Algorithm

Input : Table of cosets, parity-check matrix H, and received vector $\boldsymbol{r}$
Output : Decoded vector

1 $\boldsymbol{s} \leftarrow H\boldsymbol{r}^\top$
2 Look up the coset leader corresponding to $\boldsymbol{s}$, say ℓ.
3 **return** $\boldsymbol{r} - \ell$

EXAMPLE 3.7.2: Syndrome Decoding

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}_{2\times 5}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{3\times 5}$$

Table 3.1: Table of Cosets

$$C + 00000 = \{00000, 10111, 01110, 11001\} \qquad C + 00010 = \{00010, 10101, 01100, 11011\}$$
$$C + 10000 = \{10000, 00111, 11110, 01001\} \qquad C + 00001 = \{00001, 10110, 01111, 11000\}$$
$$C + 01000 = \{01000, 11111, 00000, 10001\} \qquad C + 00011 = \{00011, 10100, 01101, 11010\}$$
$$C + 00100 = \{00100, 10011, 01010, 11101\} \qquad C + 11100 = \{11100, 01011, 10010, 00101\}$$

There are $q^{n-k} = 2^{5-2} = 2^3 = 8$ cosets in total.

Coset Leaders	Syndromes
00000	000
10000	111
01000	110
00100	100
00010	010
00001	001
00011	011
10010	101

Suppose $\boldsymbol{r} = (10111)$ is received. Decode $\boldsymbol{r}$.
Solution.
Compute $\boldsymbol{s} = H\boldsymbol{r}^\top = (000)^\top$.
The closest leader corresponding to $\boldsymbol{s} = (000)$ is $\ell = (00000)$.
Thus, we get the decoded vector $\boldsymbol{r} - \ell = (10111)$.

REMARK 3.7.3

Syndrome decoding is *not* efficient in general since the syndrome table is exponentially large:
For an (n, k)-binary code, the syndrome table has size

$$2^{n-k}[n + (n - k)] = 2^{n-k}(2n - k) \text{ bits}$$

Actually, $2^{n-k}n$ bits, since the table can be sorted by syndrome, and then the syndromes do not need to be stored.

Chapter 4

Some Special Linear Codes

2020-02-07

DEFINITION 4.0.1: Self-orthogonal

A linear code C is **self-orthogonal** if $C \subseteq C^{\perp}$.

DEFINITION 4.0.2: Self-dual

A linear code C is **self-dual** if $C = C^{\perp}$.

For a binary (n, k)-code C, the syndrome table has size $2^{n-k} \times n$ which is exponentially large.

Goal: Design decoding algorithm which require very little space.

EXAMPLE 4.0.3

Use only the PCM H which is $(n-k) \times n$ bits.

4.1 The Binary Golay Code C23 (1949)

DEFINITION 4.1.1: C_{23}

Let

$$\hat{B} = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&0&1&1&1&0&0&0&1&0 \\ 1&0&1&1&1&0&0&0&1&0&1 \\ 0&1&1&1&0&0&0&1&0&1&1 \\ 1&1&1&0&0&0&1&0&1&1&0 \\ 1&1&0&0&0&1&0&1&1&0&1 \\ 1&0&0&0&1&0&1&1&0&1&1 \\ 0&0&0&1&0&1&1&0&1&1&1 \\ 0&0&1&0&1&1&0&1&1&1&0 \\ 0&1&0&1&1&0&1&1&1&0&0 \\ 1&0&1&1&0&1&1&1&0&0&0 \\ 0&1&1&0&1&1&1&0&0&0&1 \end{bmatrix}_{12\times 11}$$

Then, $\hat{G} = \left[I_{12} \mid \hat{B} \right]_{12\times 23}$ is a generator matrix for a $(23, 12)$-binary code called $\boldsymbol{C_{23}}$.

Note: In $\hat{B}$,

- R_1 in only contains 1's.
- R_3 to R_{12} are left cyclic shifts of R_2.

THEOREM 4.1.2

Facts:

1. $d(C_{23}) = 7$.
2. C_{23} *is perfect.*

Proof of Theorem 4.1.2

We know that $e = 3$, so $2^{12}\left[\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}\right] = 2^{23}$.

4.2 The Extended Golay Code C24

DEFINITION 4.2.1: C_{24}

Let

$$B = \left[\begin{array}{c|c} 0 & \\ \mathbf{1} & \hat{B} \end{array}\right]_{12\times 12}$$

where $\mathbf{1}$ is the column vector $(\underbrace{1, \ldots, 1}_{11 \text{ times}})^\top$.

Then, $G = \left[I_{12} \mid B \right]_{12\times 24}$ is a generator matrix for a $(24, 12)$-binary code called $\boldsymbol{C_{24}}$.

Notes:

(1) C_{24} is a $(24, 12, 8)$-binary code.

(2) $GG^\top = 0$.

(3) $C_{24} \subseteq C_{24}^\perp$, so C_{24} is a self-orthogonal code.

(4) $\dim C_{24} = 12 = \dim C_{24}^\perp$, so $C_{24} = C_{24}^\perp$, therefore C_{24} is a self-dual code.

(5) B is symmetric.

(6) A PCM for C_{24} is $H = [-B^\top \mid I_{12}] = [B \mid I_{12}]$.

(7) $C_{24} = C_{24}^\perp$, thus H is also a GM and PCM for C_{24}.

(8) G is also a GM and PCM for $C_{24}^\perp$.

Decoding Algorithm for C24

Compute a syndrome of $\boldsymbol{r}$. Find a vector $\boldsymbol{e}$ with $w(\boldsymbol{e}) \leqslant 3$, that has the same syndrome as $\boldsymbol{r}$. If no such $\boldsymbol{e}$ exists, then reject $\boldsymbol{r}$, otherwise decode $\boldsymbol{r}$ to $\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e}$.

Let $\boldsymbol{r} = (\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$. There are five (not mutually exclusive) cases to consider. In the event that $w(\boldsymbol{e}) \leqslant 3$,

(A) $w(\boldsymbol{e}_1) = 0$, $w(\boldsymbol{e}_2) = 0$

(B) $1 \leqslant w(\boldsymbol{e}_1) \leqslant 3$, $w(\boldsymbol{e}_2) = 0$

(C) $w(\boldsymbol{e}_1) = 1$ or 2, $w(\boldsymbol{e}_2) = 1$

(D) $w(\boldsymbol{e}_1) = 0$, $1 \leqslant w(\boldsymbol{e}_2) \leqslant 3$

(E) $w(\boldsymbol{e}_1) = 1$, $w(\boldsymbol{e}_2) = 1$ or 2

THEOREM 4.2.2

Let C be an (n, k, d)-code over $GF(q)$. Let $\boldsymbol{x} = V_n(GF(q))$ with $w(\boldsymbol{x}) \leqslant \lfloor \frac{d-1}{2} \rfloor$. Then $\boldsymbol{x}$ is the unique vector of minimum weight in the coset of C containing $\boldsymbol{x}$ (so, it must be a coset leader).

Proof of Theorem 4.2.2

Suppose for a contradiction that $\boldsymbol{y}$ is a vector in the same coset of C as $\boldsymbol{x}$ with $\boldsymbol{y} \neq \boldsymbol{x}$ and

$$w(\boldsymbol{y}) \leqslant w(\boldsymbol{x}) \leqslant \left\lfloor \frac{d-1}{2} \right\rfloor$$

Then, $\boldsymbol{y} - \boldsymbol{x} \neq \boldsymbol{0}$, $\boldsymbol{x} \equiv \boldsymbol{y} \pmod{C} \iff (\boldsymbol{x} - \boldsymbol{y}) \in C$. Now,

$$\begin{aligned} w(\boldsymbol{x} - \boldsymbol{y}) = w(\boldsymbol{x} + (-\boldsymbol{y})) &\leqslant w(\boldsymbol{x}) + w(-\boldsymbol{y}) \\ &= w(\boldsymbol{x}) + w(\boldsymbol{y}) \\ &\leqslant \left\lfloor \frac{d-1}{2} \right\rfloor + \left\lfloor \frac{d-1}{2} \right\rfloor \\ &\leqslant d - 1 \end{aligned}$$

contradicting $d(C) = d$.

2020-02-12

Algorithm 3: Decoding Algorithm for C_{24}

Input : $G = [\, I_{12} \mid B \,] = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{24})^\top$, and $\tilde{G} = [\, B \mid I_{12} \,] = (\tilde{\boldsymbol{g}}_1, \ldots, \tilde{\boldsymbol{g}}_{24})^\top$, and received vector $\boldsymbol{r} = (\boldsymbol{x}, \boldsymbol{y})$

Output : Decoded vector

$\boldsymbol{s}_1 \leftarrow G\boldsymbol{r}^\top$
if $\boldsymbol{s}_1 = \boldsymbol{0}$ **then**
  return $\boldsymbol{r}$
if $w(\boldsymbol{s}_1) \leqslant 3$ **then**
  for i **to** 12 **do**
    $\boldsymbol{x}' \leftarrow$ corrected $\boldsymbol{x}$ in each position corresponding to 1's in $\boldsymbol{s}_1$
    return $(\boldsymbol{x}', \boldsymbol{y})$
for $i \leftarrow 0$ **to** 24 **do**
  if $\boldsymbol{g}_i$ *differs in position* j *or positions* j *and* k *from* $\boldsymbol{s}_1$ **then**
    $\boldsymbol{x}' \leftarrow (x_1, \ldots, x_{12})$ where $x_j \leftarrow \overline{x}_j$ or $x_j \leftarrow \overline{x}_j$ and $x_k \leftarrow \overline{x}_k$
    $\boldsymbol{y}' \leftarrow (y_1, \ldots, y_{12})$ where $y_i \leftarrow \overline{y}_i$
    return $(\boldsymbol{x}', \boldsymbol{y}')$
$\boldsymbol{s}_2 \leftarrow \tilde{G}\boldsymbol{r}^\top$
if $w(\boldsymbol{s}_2) \leqslant 3$ **then**
  $\boldsymbol{y}' \leftarrow$ corrected $\boldsymbol{y}$ in each position corresponding to 1's in $\boldsymbol{s}_2$
  return $(\boldsymbol{x}, \boldsymbol{y}')$
for $i \leftarrow 0$ **to** 24 **do**
  if $\tilde{\boldsymbol{g}}_i$ *differs in position* j *or positions* j *and* k *from* $\boldsymbol{s}_2$ **then**
    $\boldsymbol{y}' \leftarrow (y_1, \ldots, y_{12})$ where $y_j \leftarrow \overline{y}_j$ or $y_j \leftarrow \overline{y}_j$ and $y_k \leftarrow \overline{y}_k$
    $\boldsymbol{x}' \leftarrow (x_1, \ldots, x_{12})$ where $x_i \leftarrow \overline{x}_i$
    return $(\boldsymbol{x}', \boldsymbol{y}')$
return

EXAMPLE 4.2.3: Decoding Algorithm for C24

1. Decode $\boldsymbol{r} = (1000\ 1000\ 0000\ 1001\ 0001\ 1101)$.
   Solution. Compute $\boldsymbol{s}_1 = [\, I_{12} \mid B \,]\, \boldsymbol{r}^\top = (0100\ 1000\ 0000)$. Since $w(\boldsymbol{s}_1) \leqslant 3$, we set $\boldsymbol{e} = (\boldsymbol{s}_1, 0)$ and decode $\boldsymbol{r}$ to
   $$\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = (1100\ 0000\ 0000\ 1001\ 0001\ 1101)$$
2. Decode $\boldsymbol{r} = (1000\ 0010\ 0000\ 1000\ 1101\ 0010)$.
   Solution. Compute $\boldsymbol{s}_1 = [\, I_{12} \mid B \,]\, \boldsymbol{r}^\top = (1011\ 1110\ 1011)$. Note that $w(\boldsymbol{s}_1) > 3$. Comparing $\boldsymbol{s}_1$ with the rows of B, we see that $\boldsymbol{s}_1$ differs in positions 6 and 7 from row 4 of B. Hence, we set $\boldsymbol{e} = (0000\ 0110\ 0000\ 0001\ 0000\ 0000)$ and decode $\boldsymbol{r}$ to
   $$\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = (1000\ 0100\ 0000\ 1001\ 1101\ 0010)$$

Note: In both examples we should check out answers by verifying that $H\boldsymbol{c}^\top = \boldsymbol{0}$ (i.e., $\boldsymbol{c}$ is indeed a codeword).

Note:

(1) If $w(\boldsymbol{e}) \leqslant 3$, then the algorithm makes the correct decision.

(2) No storage is needed:
$$\boldsymbol{s}_1 = [\, I_{12} \mid B \,]\, \boldsymbol{r}^\top = [\, I_{12} \mid B \,] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{x} + B\boldsymbol{y}$$

where B is a left cyclic shift of the first row.

(3) The algorithm is very simple and efficient for hardware.

Reliability of C24

- $p =$ symbol error probability.
- $C = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_M\}$.
- $w_i =$ probability that the decoding algorithm makes an incorrect decision if $\boldsymbol{c}_i$ is sent.
- $P_C = \frac{1}{M} \sum_{i=1}^{M} w_i$ error probability of C.
- $1 - P_C =$ reliability of C (correct decision).

p	$(1-p)^{12}$	$1 - P_{C_{24}}$	$1 - P_T$	$1 - P_H$
0.1	0.28243	0.785738	0.71121	0.549043
0.01	0.886385	0.999909	0.99643	0.99037
0.001	0.988066	≈ 1	0.999964	0.999896
Rate	1	$1/2 = 0.5$	$1/3 = 0.3\overline{3}$	$11/15 = 0.7\overline{3}$

(1) If no source is used, then the reliability for 12-bit messages is

$$(1-p)^{12}$$

(2) C_{24}

$$1 - P_{C_{24}} = \left[(1-p)^{24} + \binom{24}{1} p(1-p)^{23} + \binom{24}{2} p^2(1-p)^{22} + \binom{24}{3} p^3(1-p)^{21}\right]$$

(3) Triplication Code T

$$1 - P_T = \left[(1-p)^3 + 3p(1-p)^2\right]^{12}$$

(4) $(15, 11)$-binary Hamming Code

$$1 - P_H = (1-p)^{15} + 15p(1-p)^{14}$$

Chapter 5

Cyclic Codes

2020-02-14 ♡

5.1 Introduction

DEFINITION 5.1.1: Cyclic subspace

A subspace S of $V_n(F)$ is a **cyclic subspace** if $(a_0, a_1, \dots, a_{n-1}) \in S \implies (a_{n-1}, a_0, \dots, a_{n-2}) \in S$.

DEFINITION 5.1.2: Cyclic code

A **cyclic code** is a cyclic subspace of $V_n(F)$.

5.2 Rings and Ideals

Let $R = F[x]/(x^n - 1)$. We write

$$\underbrace{(a_0, a_1, \dots, a_{n-1})}_{\in V_n(F)} \longleftrightarrow \underbrace{a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}}_{\in R}$$

That is, there is an isomorphism between $V_n(F)$ and R.

- Addition is preserved: $\boldsymbol{a} + \boldsymbol{b} \longleftrightarrow a(x) + b(x)$
- Scalar multiplication is preserved: $\lambda \boldsymbol{a} \longleftrightarrow \lambda a(x)$

Why choose $x^n - 1$?

Let $\boldsymbol{a} = (a_0, \dots, a_{n-1}) \in V_n(F)$. Let $a(x)$ be the associated polynomial in R. Then,

$$\begin{aligned} x \cdot a(x) &= a_0 x + a_1 x^2 + \cdots + a_{n-2}x^{n-1} + a_{n-1}x^n \\ &\equiv a_{n-1} + a_0 x + \cdots + a_{n-2}x^{n-1} \pmod{x^n - 1} \\ &\longleftrightarrow (a_{n-1}, a_0, \dots, a_{n-2}) \end{aligned}$$

So, multiplying a polynomial in R by x corresponds to a right cyclic shift of the associated vector.

We'll define $\cdot : V_n(F) \times V_n(F) \to V_n(F)$ by

$$a \cdot b \longleftrightarrow a(x)b(x) \pmod{x^n - 1}$$

DEFINITION 5.2.1: Ideal

Let R be a commutative finite ring. Then, the non-empty subset I of R is an **ideal** of R if
(1) For all $a, b \in I$, $a + b \in I$
(2) For all $a \in I$, $b \in R$, $ab \in I$
$\{0\}$ and R are defined to be **trivial** ideals of R.

THEOREM 5.2.2

Let $S \subseteq V_n(F)$ be non-empty. Let I be the associated polynomials. Then S is a cyclic subspace of $V_n(F)$ if and only if I is an ideal of $R = F[x]/(x^n - 1)$.

Proof of Theorem 5.2.2

($\implies$) Suppose S is a cyclic subspace of $V_n(F)$. Since S is closed under addition, so is I. Let $a(x) \in I$, $b(x) = b_0 + \cdots + b_{n-1}x^{n-1} \in R$. Then $xa(x) \in I$ since S is a cyclic subspace. So, $x^i a(x) \in I$ for each $i \in [0, n-1]$. Also, $b_i x^i a(x) \in I$ since S is closed under scalar multiplication. Finally, $a(x)b(x) = a(x)(b_0 + \cdots + b_{n-1}x^{n-1})$ which is in I since I is closed under addition. Thus, I is an ideal.
($\impliedby$) Suppose I is an ideal of R. Since I is closed under addition, so is S. Since I is closed under multiplication by constant polynomials, S is closed under scalar multiplication. Since I is closed under multiplication by x, S is closed under (right) cyclic shifts. Thus, S is a cyclic subspace.

DEFINITION 5.2.3: Ideal generated by $g(x)$

Let $g(x) \in R$. Then $\langle g(x) \rangle = \{g(x)a(x) : a(x) \in R\}$ is an ideal of R called the **ideal generated by $g(x)$**. If I is an ideal of R, then I is a **principal** ideal if there exists a $g(x) \in I$ such that $I = \langle g(x) \rangle$. R is called the **principal ideal ring** if every ideal ring of R is principal.

THEOREM 5.2.4

$R = F[x]/(x^n - 1)$ is a principal ideal ring.

Proof of Theorem 5.2.4

Let I be an ideal of R.
Suppose $I = \{0\}$, then $I = \langle 0 \rangle$ is principal.
Suppose $I \neq 0$. Let $g(x)$ be a polynomial of smallest degree in I. Let $a(x) \in I$. Long division gives

$$a(x) = \ell(x)g(x) + r(x)$$

where $\ell, r \in F[x]$ and $\deg(r) < \deg(g)$, but $\ell(x)g(x) \in I$ since I is closed under multiplication by R and $a(x) = \ell(x)g(x) \in I$. Therefore, $r(x) \in I$. Since $\deg(r) < \deg(g)$, we must have $r(x) = 0$ (since we define $\deg(0) = -\infty$). Hence, $a(x) = \ell(x)g(x)$. Therefore, $I = \langle g(x) \rangle$. Thus, R is a principal ideal ring.

5.3 Ideals and Cyclic Subspaces

DEFINITION 5.3.1: Monic polynomial

A **monic polynomial** $g(x)$ is a single-variable polynomial in which the non-zero coefficient of the highest degree of x is 1. That is,

$$g(x) = c_0 + \cdots + c_{\ell-1}x^{\ell-1} + x^\ell$$

for some constants c_i where $i \in [\ell - 1, 1]$.

If $I \neq \{0\}$, then we took $g(x) = a$ non-zero polynomial of smallest degree in I. Note, we can take $g(x)$ to be monic. If $g(x)$ is not monic, say

$$g(x) = c_0 + \cdots + c_\ell x^\ell$$

where $c_\ell \neq 0, 1$, then

$$c_\ell^{-1} g(x) = c_\ell^{-1} g_0 + \cdots x^\ell$$

is monic and is also in I. We'll call this process **making $g(x)$ monic**.

DEFINITION 5.3.2: Generator polynomial of I

Let I be an ideal in $R = F[x]/(x^n - 1)$.
The **generator polynomial of I** is:
(1) $x^n - 1$ since $x^n - 1 \equiv 0 \pmod{x^n - 1}$ when $I = \{0\}$.
(2) **The** monic polynomial of least degree in I when $I \neq \{0\}$.

THEOREM 5.3.3

Let I be a non-zero ideal in $R = F[x]/(x^n - 1)$.
*(1) There is a **unique** monic polynomial g(x) of smallest degree in I.*
(2) $g(x) \mid (x^n - 1)$.

Proof of Theorem 5.3.3

(1) Suppose there exists two monic polynomials $g(x)$ and $h(x)$ of the same smallest degree in I. Then, $g(x) - h(x) \in I$ and $\deg(g - h) < \deg(g)$. Hence, we must have $g - h = 0$, so $g = h$.
(2) We can write

$$x^n - 1 = \ell(x)g(x) + r(x)$$

where $\ell, r \in F[x]$ and $\deg(r) < \deg(g)$. Then,

$$0 \equiv \ell(x)g(x) + r(x) \pmod{x^n - 1} \iff r(x) \equiv -\ell(x)g(x) \pmod{x^n - 1}$$

Since $\langle g(x) \rangle = I$, we must have $r(x) \in I$. Hence, $\deg(r) < \deg(g)$ so we must have $r(x) = 0$. Thus,

$$g(x) \mid (x^n - 1)$$

THEOREM 5.3.4

*Let $h(x)$ be a monic divisor of $x^n - 1$ in $F[x]$. Then, **the** generator polynomial of $\langle h(x) \rangle$ is $h(x)$.*

Proof of Theorem 5.3.4

If $h(x) = x^n - 1$, then $I = \{0\}$ and by definition, its generator polynomial is $x^n - 1$.
If $\deg(h) < n$, then $I \neq \{0\}$. Let $g(x)$ be **the** monic polynomial of smallest degree in I. Since g is a generator of I, we can write

$$g(x) \equiv a(x)h(x) \pmod{x^n - 1} \implies g(x) = a(x)h(x) + \ell(x)(x^n - 1)$$

for some $\ell \in F[x]$. Since $h \mid (x^n - 1)$, and $h \mid ah$, we have $h \mid g$. So, $\deg(h) \leqslant \deg(g)$ since g is a monic polynomial of smallest degree in I, we must have $\deg(g) \leqslant \deg(h)$, so $\deg(g) = \deg(h)$. Since g and h are both monic, we have $g = h$.

COROLLARY 5.3.5

There is a 1–1 correspondence between monic divisors of $x^n - 1$ in $F[x]$ and ideals in R. There is a 1–1 correspondence between monic divisors of $x^n - 1$ in $F[x]$ and cyclic subspaces of $V_n(F)$.

EXAMPLE 5.3.6

Find all cyclic subspaces of $V_3(\mathbb{Z}_2)$.
Solution. The complete factorization of $x^3 - 1$ over $\mathbb{Z}_2$ is

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

Monic divisor of $x^3 - 1$	$\langle g_i(x) \rangle$	$\dim\langle g_i(x) \rangle$
$g_1(x) = 1$	$\{000, 001, \ldots, 111\}$	3
$g_2(x) = 1 + x$	$\{000, 110, 001, 101\}$	2
$g_3(x) = 1 + x + x^2$	$\{000, 111\}$	1
$g_4(x) = 1 + x^3$	$\{0\}$	0

2020-02-26

Midterm review session.

2020-02-28

$$\begin{array}{ccc} V_n(F) & \longleftrightarrow & R = F[x]/(x^n - 1) \\ \boldsymbol{a} = (a_0, a_1 \ldots, a_{n-1}) \in V_n(F) & \longleftrightarrow & a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in R \\ C: \text{ cyclic subspace, with } \dim(C) = k & \longleftrightarrow & I: \text{ ideal in } R \\ & & g(x) \text{ with } \deg(g) = n - k, \text{ if GM for } C \text{ in terms of } g(x) \\ \text{Encoding: } \boldsymbol{m}G & \longleftrightarrow & m(x)g(x) \\ C^{\perp} & \longleftrightarrow & h^*(x) \\ \text{PCM for } C: H & \longleftrightarrow & s(x) \equiv r(x) \pmod{g(x)} \end{array}$$

To find $h^*(x)$, we need $h(x) = (x^n - 1)/(g(x))$ where $\deg(h) = k$. Then, we find the reciprocal polynomial $h_R(x)$, and we make it monic to obtain $h^*(x)$.

Note: We do not know the distance of C, but we can use a BCH code and specifically select $g(x)$ to give a lower bound on $d(C)$.

LEMMA 5.3.7

Let $g(x)$ be a monic divisor with $\deg(g) = n - k$ of $x^n - 1$ in $F[x]$. In fact,

$$\langle g(x) \rangle = \{g(x)\overline{a}(x) : \deg(\overline{a}) < k\}$$

Proof of Lemma 5.3.7

Let $h(x) = g(x)a(x) \pmod{x^n - 1}$ for some $a(x)$ where $\deg(a) < n$. So,

$$h(x) - g(x)a(x) = \ell(x)(x^n - 1)$$

for some $\ell \in F[x]$. Therefore, $g \mid h$. So, $h(x) = g(x)\overline{a}(x)$, for some $\overline{a} \in F[x]$ with $\deg(\overline{a}) \leqslant k - 1$.

THEOREM 5.3.8

Let $g(x)$ be a monic divisor of $x^n - 1$ with $\deg(g) = n - k$ of $x^n - 1$ in $F[x]$. Then, the cyclic code C generated by $g(x)$ has dimension k.

Proof of Theorem 5.3.8

We'll show that

$$B = \{g(x), xg(x), \dots, x^{k-1}g(x)\}$$

is a basis of C.
We first show B is linearly independent. Suppose

$$\lambda_0 g(x) + \lambda_1 xg(x) + \cdots + \lambda_{k-1}x^{k-1}g(x) = 0$$

where $\lambda_i \in F$ for each $i \in [0, k-1]$. The coefficient x^{n-1} in the LHS is λ_{k-1}. The coefficient of x^{n-1} in the RHS is 0. Hence, $\lambda_{k-1} = 0$. Similarly,

$$\lambda_0 = \lambda_1 = \cdots = \lambda_{k-2} = 0$$

Thus, B is linearly independent.
We now show B spans C. Let $h(x) \in \langle g(x) \rangle$. By Lemma, we can write

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

for some $a \in F[x]$ where $\deg(g) = n - k$ and $\deg(a) \leqslant k - 1$. Let

$$a(x) = \sum_{i=0}^{k-1} a_i x_i$$

where $a_i \in F$ for each $i \in [0, k-1]$. Then,

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

Thus, $\dim(C) = k$.

5.4 Generator Matrices and Parity-Check Matrices

Therefore, a generator matrix for C is:

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}_{k \times n} = \begin{bmatrix} g(x) & 0 & \cdots & 0 & 0 \\ 0 & xg(x) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & x^{k-2}g(x) & 0 \\ 0 & 0 & \cdots & 0 & x^{k-1}g(x) \end{bmatrix}_{k \times n}$$

Note: G is a non-systematic generator matrix for C.

Encoding

$$\begin{aligned}
\boldsymbol{c} &= \boldsymbol{m}G \\
&= (m_0, \dots, m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix} \\
&= m_0 g(x) + m_{k-1}x^{k-1}g(x) \\
&= g(x)(m_0 + \cdots + m_{k-1}x^{k-1}) \\
&\implies c(x) = m(x)g(x)
\end{aligned}$$

EXAMPLE 5.4.1

Construct a cyclic $(7, 4)$-code over $\mathbb{Z}_2$.
Solution. We need a monic divisor of degree 3 of $x^7 - 1$ in $\mathbb{Z}_2[x]$. Using Table 3 on page 157:

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

Let's take $g(x) = 1 + x + x^3$. Then, $\langle g(x) \rangle$ is a $(7, 4)$-cyclic code over $\mathbb{Z}_2$. A generator matrix for C is:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{4\times 7}$$

Encode $\boldsymbol{m} = (1011)$.
Solution.

$$\boldsymbol{c} = \boldsymbol{m}G = (1111111)$$
$$\implies c(x) = m(x)g(x) = (1 + x + x^3)(1 + x + x^3) = (1 + x + \cdots + x^6) = \boldsymbol{c}$$

2020-03-02

Let C be an (n, k)-cyclic cover over F with generator polynomial $g(x)$. Let

$$g(x) = \underbrace{g_0}_{\neq 0} + g_1 x + \cdots + \underbrace{g_{n-k}x^{n-k}}_{=1} + \underbrace{g_{n-k+1}x^{n-k+1} + \cdots + g_{n-1}x^{n-1}}_{=0}$$

Let

$$h(x) = (x^n - 1)/(g(x)) = h_0 + h_1 x + \cdots + h_{k-1}x^{k-1} + h_k x^k + \cdots + h_{n-1}x^{n+0}$$

Let $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$. We know that

$$a(x) = g(x)h(x) \pmod{x^n - 1} \qquad (\star)$$

Note: $a(x) = 0$. Equating coefficients of x^i for each $i \in [0, n-1]$ of $(\star)$:

$$a_i = 0 = g_0 h_i + g_1 h_{i-1} + \cdots + g_i h_0 + g_{i+1}h_{n-1} + g_{i+1}h_{n-2} + \cdots + g_{n-1}h_{i-1}$$

Let $g = (g_0, \dots, g_{n-1})$, $\overline{h} = (h_{n-1}, \dots, h_0)$. Then, g is orthogonal to $\overline{h}$ and all the cyclic shifts of $\overline{h}$. Every cyclic shift of g is orthogonal to every click shift of $\overline{h}$.

Recall: A generator matrix for C is:

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} \end{bmatrix}_{k\times n}$$

Consider

$$H = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_k & h_{k-1} & h_{k-2} & \cdots & h_1 & h_0 \end{bmatrix}_{(n-k)\times n}$$

We have observed $GH^\top = 0$. Let C' be the code spanned by the rows of H. Then, $C' \subseteq C^\perp$. But, $\text{rank}(H) = n-k$ (since $h_k = 1$). So, $\dim(C') = n-k$, hence we have $C' = C^\perp$. Thus, H is a PCM for C.

DEFINITION 5.4.2: Reciprocal of h

Let $h(x) = h(x) = h_0 + h_1 x + \cdots h_k x^k$ be a degree k polynomial. The **reciprocal of h** is

$$h_R(x) = h_k x^0 + \cdots + h_1 x^{k-1} + h_0 x^k$$

Note:

- $h_R(x) = x^k h\left(\frac{1}{x}\right)$
- If $h_0 \neq 0$, then $h^*(x) = h_0^{-1} h_R(x)$.

THEOREM 5.4.3

If C is an (n,k)-cyclic code, then $C^\perp$ is an $(n, n-k)$ cyclic code.

Proof of Theorem 5.4.3

$$\begin{aligned} & g(x)h(x) = x^n - 1 \\ \implies & g\left(\frac{1}{x}\right) h\left(\frac{1}{x}\right) = \left(\frac{1}{x^n} - 1\right) \\ \implies & x^{n-k} g\left(\frac{1}{x}\right) \left(x^k h\left(\frac{1}{x}\right)\right) = (1 - x^n) \\ \implies & g_R(x) h_R(x) = -(x^n - 1) \\ \implies & h_R(x) \mid (x^n - 1) \end{aligned}$$

So, $h_R(x)$ is a degree k divisor of $x^n - 1$. Hence, the matrix H is a generator matrix for the cyclic code generated by $h^*(x)$. Thus, $C^\perp$ is cyclic with generator polynomial $h^*(x)$.

5.5 Syndromes and Simple Decoding Procedures

$s = H\boldsymbol{r}^\top$. Let's find a more convenient PCM for C.

(i) Find a generator matrix for C of the form $\begin{bmatrix} R & \mid & I_k \end{bmatrix}_{k\times n}$ is (essentially systematic). For each $i \in [0, k-1]$, long division gives:

$$x^{n-k+i} = \underbrace{\ell_i(x) g(x)}_{\deg = n-k} + \underbrace{r_i(x)}_{\deg \leqslant n-k-1}$$

Then, $-r_i(x) + x^{n-k+i} = \ell_i(x)g(x) \in C$. Let

$$G = \begin{bmatrix} -r_0(x) + x^{n-k} \\ -r_1(x) + x^{n-k+1} \\ \vdots \\ -r_{k-1}(x) + x^{n-1} \end{bmatrix} = \left[\, R \mid I_k \,\right]_{k\times n}$$

G has rank $= k$, so G is a GM for C.

(ii) Construct a PCM for C.

This is $H = \left[\, I_{n-k} \mid -R^\top \,\right]_{(n-k)\times n}$. Then, $H\boldsymbol{r}^\top = r(x) \pmod{g(x)}$.

2020-03-04

Recall: Let C be an (n,k)-cyclic code over $GF(q)$ with generator polynomial $g(x)$. One generator matrix for C is:

$$\begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}_{k\times n}$$

One PCM for C is:

$$H = \begin{bmatrix} h^*(x) \\ xh^*(x) \\ \vdots \\ x^{n-k-1}h^*(x) \end{bmatrix}_{(n-k)\times n}$$

Another generator matrix for C is:

$$G = \left[\, R \mid I_k \,\right] = \left[\begin{array}{c} -r_0(x) \\ -r_1(x) \\ \vdots \\ -r_{k-2}(x) \\ -r_{k-1}(x) \end{array}\middle|\; I_k \right]$$

where $x^{n-k+i} = \ell_i(x)g(x) + r_i(x) \implies -r_i(x) + x^{n-k+i} = \ell_i(x)g(x)$ for each $i \in [0, k-1]$. Then, another PCM for C is: $H = \left[\, I_{n-k} \mid -R^\top \,\right]_{(n-k)\times n}$. So,

$$H^\top = \begin{bmatrix} I_{n-k} \\ -R \end{bmatrix}_{n\times(n-k)} = \begin{bmatrix} x^0 \pmod{g(x)} \\ x \pmod{g(x)} \\ \vdots \\ x^{n-k-1} \pmod{g(x)} \\ x^{n-k} \pmod{g(x)} \\ x^{n-k+1} \pmod{g(x)} \\ x^{n-1} \pmod{g(x)} \end{bmatrix}$$

Hence, if $\boldsymbol{r} = (r_0, r_1, \ldots, r_{n-1}) \in V_{n-1}(F)$, then

$$\begin{aligned} \boldsymbol{s} &= H\boldsymbol{r}^\top \\ &= (r_0x^0 \pmod{g(x)}) + \cdots + (r_{n-1}x^{n-1} \pmod{g(x)}) \\ &= (r_0x^0 + r_1x + \cdots + r_{n-1}x^{n-1}) \pmod{g(x)} \\ &= r(x) \pmod{g(x)} \end{aligned}$$

THEOREM 5.5.1

Let C be a cyclic code with generator polynomial $g(x)$, and $\boldsymbol{r} \in V_n(F)$. Then, the syndrome of $\boldsymbol{r}$ with respect to the previous PCM is:

$$s(x) = r(x) \pmod{g(x)}$$

EXAMPLE 5.5.2

$g(x) = 1 + x + x^2 + x^3 + x^6$ is the generator polynomial for a $(15, 9)$-binary cyclic code. Check $g(x) \mid (x^{15} - 1)$ over $GF(2)$. Compute the syndrome of $\boldsymbol{r} = (1110\ 1110\ 1100\ 000)$.

Solution. Long division of $(x^9+x^8+x^6+x^5+x^4+x^2+x+1)/(x^6+x^3+x^2+x+1)$ gives x^5+x^4+x+1 as the remainder. Thus,

$$s(x) = 1 + x + x^4 + x^5 \implies \boldsymbol{s} = (110011)$$

REMARK 5.5.3

Given the syndrome $\boldsymbol{s}$ of $\boldsymbol{r}$, the syndromes of cyclic shifts of $\boldsymbol{r}$ can be easily computed.

THEOREM 5.5.4

Let $\boldsymbol{r} \in V_n(F)$, and $s(x) \equiv r(x) \pmod{g(x)}$ where $s(x) = s_0 + xs_1 + \cdots + s_{n-k-1}x^{n-k-1}$. Then the syndrome of $xr(x)$ is:

(i) $xs(x)$, if $s_{n-k-1} = 0$.

(ii) $xs(x) - s_{n-k-1}g(x)$, if $s_{n-k-1} \neq 0$.

Proof of Theorem 5.5.4

We have

$$r(x) = \ell(x)g(x) + s(x)$$

Multiply by x,

$$xr(x) = x\ell(x)g(x) + xs(x)$$

- **Case 1**: If $s_{n-k-1} = 0$, then $\deg(s) \leqslant n - k - 2$, so $\deg(xs(x)) \leqslant n - k - 1$. So, $xs(x)$ is **the** remainder upon dividing $xr(x)$ by $g(x)$. So, $xs(x)$ is **the** syndrome of $r(x)$.
- **Case 2**: If $s_{n-k-1} \neq 0$, then $\deg(s) = n - k - 1$. Then

$$xr(x) = x\ell(x)g(x) + xs(x) + s_{n-k-1}g(x) - s_{n-k-1}g(x)$$

$$\implies xr(x) = (x\ell(x) + s_{n-k-1})g(x) + (xs(x) - s_{n-k-1}g(x))$$

Now,

$$xs(x) - s_{n-k-1} = (s_0 + \cdots + s_{n-k-1}x^{n-k}) - (\cdots + s_{n-k-1}x^{n-k}) = xr(x)$$

So, $xs(x) - s_{n-k-1}g(x)$ is **the** syndrome of $xr(x)$.

2020-03-06

5.6 Burst Error Correcting

"Cyclic codes are good for (cyclic) burst error correcting"

Suppose we have a $C : (n, k, d)$ code, with $e = \lfloor \frac{d-1}{2} \rfloor = 5$. In practice, errors typically happen in bursts (not spread out). We expect typically one burst per codeword, or bursts to carry through two codewords.

DEFINITION 5.6.1: Cyclic burst of length e

Let $\boldsymbol{e} \in V_n(F)$. The **cyclic burst of length e** is the length of the smallest cyclic block that contain all the non-zero entries of $\boldsymbol{e}$.

EXAMPLE 5.6.2

$\boldsymbol{e} = \mathbf{011}000001$ has cyclic burst length 4.

DEFINITION 5.6.3: Cyclic burst error of length t

We say $\boldsymbol{e}$ is a **cyclic burst error of length t** if its cyclic burst length is t.

DEFINITION 5.6.4: t-cyclic burst error correcting code

A linear code C is a **t-cyclic burst error correcting code** if every cyclic burst error of length at most t lies in a unique coset of C. The largest such t is called the **cyclic burst error capability of C**.

EXAMPLE 5.6.5

$g(x) = 1+x+x^2+x^3+x^6$ generates a $(15, 9)$-binary cyclic code C that is a 3-cyclic burst error correcting code.

$d(C) \leqslant 5$, so $e \leqslant 2$. We verify this by checking that each cyclic burst of length $\leqslant 3$ has a unique syndrome.

Cyclic burst errors	Syndromes	Notes
0	000000	
x^0	100000	
x^1	010000	
x^2	001000	
x^3	000100	
x^4	000010	
x^5	000001	
x^6	111100	$x^6 + g(x) \iff (0000001) + (1111001)$
x^7	011110	
x^8	001111	
x^9	111011	$x^9 + g(x) \iff (0001111) + (1111001)$
x^{10}	100001	$x^{10} + g(x) \iff (0111011) + (1111001)$
x^{11}	101100	$x^{11} + g(x) \iff (0100001) + (1111001)$
x^{12}	010110	
x^{13}	001011	
x^{14}	111001	$x^{14} + g(x) \iff (0001011) + (1111001)$
$1 + x$	110000	
$x(1 + x)$	011000	
$\vdots$	$\vdots$	
$x^{14}(1 + x)$	011001	
$1 + x + x^2$	111000	
$x(1 + x + x^2)$	011100	
$\vdots$	$\vdots$	
$x^{14}(1 + x + x^2)$	001001	
$1 + x^2$	101000	
$x(1 + x^2)$	010100	
$\vdots$	$\vdots$	
$x^{14}(1 + x^2)$	101001	

The number of cyclic bursts of length $\leqslant 3$ is 61. The number of syndromes is 64.

EXAMPLE 5.6.6

$g(x) = 1 + x^4 + x^6 + x^7 + x^8$ generates a $(15, 7)$-binary cyclic code that is 4-cyclic burst error correcting. Distance $\leqslant 5$ so $e \leqslant 2$.

Question: How to construct codes with high cyclic burst error correcting capability?

(1) Use a computer search.

(2) RS Codes.

(3) Interleaving.

THEOREM 5.6.7

Let C be an (n, k, d)-code over $GF(q)$. Let t be its cyclic burst error correcting capability.

$$\left\lfloor \frac{d-1}{2} \right\rfloor \leqslant t \leqslant n-k$$

Proof of Theorem 5.6.7

Every cyclic burst of length $\leqslant t$ has weight $\leqslant t$. Since every vector of weight $\leqslant \lfloor \frac{d-1}{2} \rfloor$ has a unique syndrome, we have $\lfloor \frac{d-1}{2} \rfloor \leqslant t$.
The number of cyclic burst errors where all the non-zero entries lie in the first t coordinate positions is q^t. Each of them has a unique coset and the total number of cosets is q^{n-k}. Thus,

$$q^t \leqslant q^{n-k} \implies t \leqslant n-k$$

EXERCISE 5.6.8

Prove that $t \leqslant \dfrac{n-k}{2}$.

5.7 Decoding Cyclic Burst Errors

Let C be a t-cyclic burst error correcting code generated by $g(x)$ which is a degree-k monic divisor of $x^n - 1$ over $GF(q)$.

Recall: A PCM for C is:

$$H = [\, I_{n-k} \mid -R^\top \,]$$

whose columns are $x^0 \pmod{g(x)}, \ldots, x^{n-1} \pmod{g(x)}$.

The syndrome of $r(x)$ is $s(x) \equiv r(x) \pmod{g(x)}$.

Idea: Suppose $\boldsymbol{e}$ is a cyclic burst of length $\leqslant t$.

Compute $\boldsymbol{s} = H\boldsymbol{r}^\top \equiv r(x) \pmod{g(x)}$.

Suppose $\boldsymbol{e} =$ x o $\cdots$ o x x x. We multiply x^3 by $\boldsymbol{e}$, so we get x x x x o $\cdots$ o.

$\boldsymbol{s} = H\boldsymbol{r}^\top = H\boldsymbol{e}^\top$.

$\boldsymbol{s}_1 = H(x\boldsymbol{r})^\top = H(x\boldsymbol{e})^\top$

$\boldsymbol{s}_2 = H(x^2\boldsymbol{r})^\top = H(x^2\boldsymbol{e})^\top$

$\boldsymbol{s}_3 = H(x^3\boldsymbol{r})^\top = H(x^3\boldsymbol{e})^\top$

2020-03-09

Recall: Let C be an (n, k) code with generator polynomial $g(x)$. Suppose C is a t-c.b.e.c.c. So, $t \leqslant n-k$.

$$H = [\, I_{n-k} \mid -R^\top \,]$$

is a PCM for C; $s(x) = r(x) \pmod{g(x)}$.

Idea: Suppose e is a cyclic burst of length at most t. Compute shifts of e, say $e_i = x^i e$ has all its non-zero entries in the first $(n-k)$ positions. Then,

$$s_i(x) = e_i(x) \pmod{g_i(x)}$$

and we can recognize such an $s_i(x)$ since it is a non-cyclic burst of length at most t. Then, $e = x^{n-i}e_i$. How do we compute $s_i(x)$? Recall, $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. So, $x^i\boldsymbol{r} = x^i\boldsymbol{c} + x^i\boldsymbol{e}$, so $x^i\boldsymbol{r}$ and $x^i\boldsymbol{e}$ have the same syndrome.

5.8 Error Trapping Decoding (For Cyclic Burst Errors)

Let $r(x)$ = received polynomial. Let $s_i(x)$ = syndrome of $x^i r(x)$ for each $i \in [1, n-1]$ where $s_0 = r(x) \pmod{g(x)}$.

Algorithm 4: Error Trapping

```
for i = 0 to n − 1 do
    Compute s_i(x) with Theorem 5.5.4.
    if s_i(x) is a non-cyclic burst of length at most t then
        e_i(x) ← (s_i(x), 0)
        e(x) ← x^{n−i} e_i(x)
        return r(x) − e(x)
return
```

EXAMPLE 5.8.1

$g(x) = 1 + x + x^2 + x^3 + x^6$ is the generator polynomial for $(15, 9)$-binary cyclic code with c.b.e.c.c 3. Decode $r = (1110\ 1110\ 1100\ 000)$.

Solution. Compute $s_0(x) = r(x) \pmod{g(x)} = x^5 + x^4 + x + 1$.

Iteration (i)	Syndrome $[s_i(x)]$
0	110011
1	100101
2	101110
3	010111
4	110111
5	100111
6	101111
7	101011
8	101001
9	101000

$$\implies \boldsymbol{e}_9 = (101000\ 000000000)$$
$$\implies \boldsymbol{e} = x^6\boldsymbol{e}_9 = (000000\ 101000\ 000)$$
$$\implies \boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = (1110\ 1100\ 0100\ 000)$$

Check: $H\boldsymbol{c}^\top = \boldsymbol{0}$ (bad) OR $g(x) \mid c(x)$ via long division.

5.9 Interleaving

Goal: Improve the c.b.e.c.c of a code.

Suppose C is an (n, k)-code with c.b.e.c.c t.

Suppose the following codewords are transmitted:

$$\begin{aligned} v_1 &= (v_{11}, v_{12}, \dots, v_{1n}) \in C \\ v_2 &= (v_{21}, v_{22}, \dots, v_{2n}) \in C \\ &\vdots \\ v_s &= (v_{s1}, v_{s2}, \dots, v_{sn}) \in C \end{aligned}$$

Suppose $v_1, \dots, v_s$ are transmitted in that order. If a cyclic burst error of length at most t occurs in any codeword, that error can be corrected.

Instead, we transmit: the **columns in order**:

$$[v_{11}, v_{21}, \dots, v_{s1}, \dots, v_{1n}, v_{2n}, \dots, v_{sn}]$$

Now, if a cyclic burst error of length at most st occurs in this (fat) codeword, this means that each original codeword suffered a cyclic error burst of length at most t.

THEOREM 5.9.1

*Suppose C is an (n, k)-cyclic code with generator polynomial $g(x)$ and cyclic burst error correcting capability t. C^*, the code obtained by **interleaving C to a depth** s is an (ns, ks)-cyclic code with generator polynomial $g^*(x) = g(x^s)$.*

2020-03-11

5.10 Minimal Polynomials

Recall that if $F = GF(p^m)$ is a finite field of characteristic p, then $\mathbb{Z}_p$ is a subfield of F, and we can view F as an m-dimensional vector space over $\mathbb{Z}_p$. More generally, for any prime power q, $GF(q)$ is a subfield of $GF(q^m)$, and we can view $GF(q^m)$ as an m-dimensional vector space over $GF(q)$.

EXAMPLE 5.10.1

$GF(2^{16})$ is:

- a 16-dimensional vector space over $GF(2)$,
- an 8-dimensional vector space over $GF(2^2)$,
- a 4-dimensional vector space over $GF(2^4)$,
- a 2-dimensional vector space over $GF(2^8)$, and
- a 1-dimensional vector space over $GF(2^{16})$.

We call $GF(q^m)$ the **extension field**, and $GF(q)$ the **subfield**. Informally, $GF(q^m)$ is the "big field," and $GF(q)$ is the "small field."

Here is the main definition in this section:

DEFINITION 5.10.2: Minimal polynomial

Let $\alpha \in GF(q^m)$. **The minimal polynomial of α over $GF(q)$**, denoted $m_\alpha(x)$, is the monic polynomial of smallest degree in $GF(q)[x]$ that has α as a root; that is, $m_\alpha(\alpha) = 0$.

REMARK 5.10.3

(1) If $m_\alpha(x) \in GF(q)[x]$ is a non-zero polynomial with $m_\alpha(\alpha)$ and c is the leading coefficient of $m_\alpha(x)$, then $m'_\alpha(x) = c^{-1}m_\alpha(x)$ is a monic polynomial in $GF(q)[x]$ with $m'_\alpha(\alpha) = 0$.
(2) More generally, multiplying a polynomial by a non-zero constant does not change the roots of the polynomial.
(3) We have $m_0(x) = x$.
(4) If $\alpha \neq 0$, let t be the order of α (recall that $t \mid (q^m - 1)$). Then, α is a root of $x^t - 1 \in GF(q)[x]$. It follows that there does indeed exist a monic polynomial of smallest degree in $GF(q)[x]$ having α as a root.

EXAMPLE 5.10.4

We found the minimal polynomial of elements in $GF(2^2) = \mathbb{Z}_2[x]/(x^2 + x + 1)$ over $GF(2)$ by trial and error:

- $m_0(y) = y$.
- $m_1(y) = y + 1$.
- $m_x(y) = y^2 + y + 1$.
- $m_{x+1}(y) = y^2 + y + 1$.

THEOREM 5.10.5

Let $\alpha \in GF(q^m)$.
(1) The minimal polynomial, $m_\alpha(x)$ of α over $GF(q)$ is unique.
(2) $m_\alpha(x)$ is irreducible over $GF(q)$.
(3) $\deg(m_\alpha) \leqslant m$.
(4) If $f(x) \in GF(q)[x]$, then, $f(\alpha) = 0$ if and only if $m_\alpha(x) \mid f(x)$.

Proof of Theorem 5.10.5 (1) to (3)

(1) Suppose there are two monic polynomials, $m_1(x)$ and $m_2(x)$, of (the same) smallest degree in $GF(q)[x]$ that have α as a root. Consider $r(x) = m_1(x) - m_2(x)$. Then,

$$r(\alpha) = m_1(\alpha) - m_2(\alpha) = 0 - 0 = 0$$

But, $\deg(r) < \deg(m_1)$, and so we conclude that $r(x) = 0$. Hence, $m_1(x) = m_2(x)$.
(2) Suppose that m_α is reducible over $GF(q)$. Then, we can write

$$m_\alpha(x) = s(x)t(x)$$

for some $s, t \in GF(q)[x]$ with $\deg(s), \deg(t) < \deg(m_\alpha)$. Then,

$$m_\alpha(\alpha) = 0 = s(\alpha)t(\alpha),$$

and hence either of $s(\alpha) = 0$ or $t(\alpha) = 0$. In either case, we have a contradiction of the minimality of $\deg(m_\alpha)$. We conclude that m_α is irreducible over $GF(q)$.
(3) Recall that $GF(q^m)$ can be viewed as an m-dimensional vector space over $GF(q)$. Thus, the $m + 1$ field elements $1, \alpha, \alpha^2, \ldots, \alpha^m$ are linearly dependent over $GF(q)$. Thus, we can write

$$a_0 + a_1\alpha + \cdots + a_m\alpha^m = 0,$$

where $a_0, a_1, \ldots, a_m \in GF(q)$, and not all are 0. Hence, α is a root of the non-zero polynomial

$$a_0 + a_1x + \cdots + a_mx^m \in GF(q)[x]$$

having degree $\leqslant m$. It follows that $\deg(m_\alpha) \leqslant m$.

2020-03-13

Proof of Theorem 5.10.5 (4)

Let $f \in GF(q)[x]$. Using the division algorithm for polynomials, we can write

$$f(x) = \ell(x)m_\alpha(x) + r(x)$$

where $\ell, r \in GF(q)[x]$ and $\deg(r) < \deg(m_\alpha)$. Now,

$$f(\alpha) = \ell(\alpha)m_\alpha(\alpha) + r(\alpha) = r(\alpha)$$

Hence,

$$f(\alpha) = 0 \iff r(\alpha) = 0 \iff r(x) = 0 \text{ since } \deg(r) < \deg(m_\alpha) \iff m_\alpha(x) \mid f(x).$$

THEOREM 5.10.6

Let $\alpha \in GF(q^m)$. Then, $\alpha \in GF(q)$ if and only if $\alpha^q = \alpha$.

Proof of Theorem 5.10.6

Since $\alpha^q = \alpha$ for all $\alpha \in GF(q)$, the elements of $GF(q)$ are roots of the polynomial $X^q - X$. Since this polynomial has degree q, it can't have any other roots in $GF(q^m)$. Thus, $\alpha \in GF(q)$ if and only if $\alpha^q = \alpha$.

DEFINITION 5.10.7: Set of conjugates of α with respect to $GF(q)$

Let $\alpha \in GF(q^m)$. Let t be the smallest positive integer such that $\alpha^{q^t} = \alpha$ (note that $t \leqslant m$). Then, the **set of conjugates of α with respect to $GF(q)$** is

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

Note that the elements of $C(\alpha)$ are distinct.

THEOREM 5.10.8

Let $\alpha \in GF(q^m)$. Then the minimal polynomial of α over $GF(q)$ is

$$\begin{aligned} m_\alpha(x) &= \prod_{\beta \in C(\alpha)} (x - \beta) \\ &= (x - \alpha)(x - \alpha^q)(x - \alpha^{q^2}) \cdots (x - \alpha^{q^{t-1}}). \end{aligned}$$

Proof of Theorem 5.10.8

(i) Clearly, $m_\alpha(x)$ is monic.
(ii) Clearly, $m_\alpha(\alpha) = 0$.
(iii) † Let $m_\alpha(x) = \sum_{i=0}^{t} m_i x^i$. The coefficients m_i are in $GF(q^m)$. We need to prove that $m_\alpha(x) \in$

$GF(q)$. Now,

$$\begin{aligned} m_\alpha(x)^q &= \prod_{\beta\in C(\alpha)} (x-\beta)^q \\ &= \prod_{\beta\in C(\alpha)} (x^q-\beta^q) \\ &= \prod_{\beta\in C(\alpha)} (x^q-\beta), \qquad \text{since } C(\alpha)=\{B^q : \beta\in C(\alpha)\} \\ &= m_\alpha(x^q) \\ &= \sum_{i=0}^{t} m_i x^{iq}. \end{aligned} \tag{1}$$

Also,

$$\begin{aligned} m_\alpha(x)^q &= \left(\sum_{i=0}^{t} m_i x^i\right)^q \\ &= \sum_{i=0}^{t} m_i^q x^{iq} \end{aligned} \tag{2}$$

Comparing coefficients of x^{iq} in (1) and (2) gives $m_i = m_i^q$ for all $i \in [0,t]$. Hence, $m_i \in GF(q)$. Thus, $m_\alpha(x) \in GF(q)[x]$.

(iv) † Let $f \in GF(q)[x]$ with $f(x) \neq 0$, and assume $f(\alpha) = 0$. Let $f(x) = \sum_{i=0}^{d} f_i x^i$. Then,

$$f(\alpha^q) = \sum_{i=0}^{d} f_i \alpha^{iq} = \left(\sum_{i=0}^{d} f_i \alpha_i\right)^q = f(\alpha)^q = 0.$$

Hence, the elements of $C(\alpha)$ are the roots of $f(x)$. Since the roots of $m_\alpha(x)$ are precisely the elements of $C(\alpha)$, we conclude that $m_\alpha(x)$ is the monic polynomial of smallest degree in $GF(q)[x]$ that has α as a root.

EXAMPLE 5.10.9: Finding the Minimal Polynomial

Consider $GF(2^4) = \mathbb{Z}_2[x]/(x^4+x+1)$. Find the minimal polynomial of $\beta = x^2+x^3$ over $\mathbb{Z}_2$. (In this example, we have $q=2$ and $m=4$)

Solution. When doing computations by hand, it will help to have a generator α of $GF(2^4)^*$, and a table of powers of α. It turns out that $\alpha = x$ is a generator as the following table shows.

$\alpha^0 = 1$	$\alpha^4 = 1+\alpha$	$\alpha^8 = 1+\alpha^2$	$\alpha^{12} = 1+\alpha+\alpha^2+\alpha^3$
$\alpha^1 = \alpha$	$\alpha^5 = \alpha+\alpha^2$	$\alpha^9 = \alpha+\alpha^3$	$\alpha^{13} = 1+\alpha^2+\alpha^3$
$\alpha^2 = \alpha^2$	$\alpha^6 = \alpha^2+\alpha^3$	$\alpha^{10} = 1+\alpha+\alpha^2$	$\alpha^{14} = 1+\alpha^3$
$\alpha^3 = \alpha^3$	$\alpha^7 = 1+\alpha+\alpha^3$	$\alpha^{11} = \alpha+\alpha^2+\alpha^3$	$\alpha^{15} = 1$

Now, $\beta = \alpha^6$. Hence, $C(\beta) = C(\alpha^6) = \{\alpha^6, \alpha^{12}, \alpha^9 = \alpha^{24}, \alpha^3 = \alpha^{18}\}$. Therefore,

$$\begin{aligned} m_\beta(y) &= (y-\alpha^6)(y-\alpha^{12})(y-\alpha^9)(y-\alpha^3) \\ &= [(y-\alpha^6)(y-\alpha^{12})][(y-\alpha^9)(y-\alpha^3)] \\ &= [y^2+(\alpha^6+\alpha^{12})y+\alpha^3][y^2+(\alpha^9+\alpha^3)y+\alpha^{12}] \\ &= [y^2+\alpha^4 y+\alpha^3][y^2+\alpha y+\alpha^{12}] \\ &= y^4+(\alpha+\alpha^4)y^3+(\alpha^{12}+\alpha^3+\alpha^5)y^2+(\alpha^{16}+\alpha^4)y+1 \\ &= y^4+y^3+y^2+y+1 \in \mathbb{Z}_2[y] \end{aligned}$$

Note that the coefficients of $m_\beta(y)$ are indeed in $GF(2)$.
Note also that we simplified terms such as $\alpha^3 + \alpha^6$ to α^2 by using the table powers of α.

2020-03-23

5.11 Finite Fields and Factoring $x^n - 1$ over $GF(q)$

Goal: Describe the factorization of $x^n - 1$ over $GF(q)$. Using this, we will see how generator polynomials $g(x)$ can be selected so that we have a lower bound on the distance of the cyclic code generated by $g(x)$; these codes are called **BCH codes**.

Let $p = \text{char}(GF(q))$. If $\gcd(n, q) \neq 1$, then write $n = \overline{n}p^\ell$, where $\ell \geqslant 1$ and $\gcd(\overline{n}, p) = 1$. Then, $x^n - 1 = (x^{\overline{n}-1})^{p^\ell}$. Without loss of generality, we shall assume that $\gcd(n, q) = 1$.

Now, let m be the smallest positive integer such that $q^m \equiv 1 \pmod{n}$; that is, $n \mid (q^m - 1)$.

Fact: m exists (beyond the scope of this course). Let α be a generator of $GF(q^m)^*$. Let $\beta = \alpha^{(q^m-1)/n} \in GF(q^m)$. Then, $\text{ord}(\beta) = n$, and the elements

$$1, \beta, \beta^2, \dots, \beta^{n-1}$$

are distinct. Furthermore,

$$(\beta^i)^n = (\beta^n)^i = 1^i = 1$$

for each $i \in [0, n-1]$. Hence,

$$1, \beta, \beta^2, \dots, \beta^{n-1}$$

are roots of $x^n - 1$; and there aren't any other roots. So,

$$x^n - 1 = (x-1)(x-\beta)(x-\beta^2)\cdots(x-\beta^{n-1})$$

is the complete factorization of $x^n - 1$ over $GF(q^m)$. However, we wanted the factorization of $x^n - 1$ over $GF(q)$.

Consider β^i for a fixed integer $i \in [0, n-1]$. Since β^i is a root of $x^n - 1$, we have $m_{\beta^i}(x) \mid (x^n - 1)$. Also, the roots of $m_{\beta^i}(x)$ are

$$C(\beta^i) = \left\{\beta^i, \beta^{iq}, \beta^{iq^2}, \dots, \beta^{iq^{t-1}}\right\}$$

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

This motivates the following definition.

DEFINITION 5.11.1: Cyclotomic coset, Set of cyclotomic cosets

Let $\gcd(n, q) = 1$ and a fixed integer $i \in [0, n-1]$. The **cyclotomic coset** of $\boldsymbol{q} \pmod{\boldsymbol{n}}$ containing $\boldsymbol{i}$ is

$$C_i = \{i, iq \pmod{n}, iq^2 \pmod{n}, \dots, iq^{t-1} \pmod{n}\}$$

where t is the smallest positive integer such that $iq^t \equiv i \pmod{n}$. Also,

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

is the **set of cyclotomic cosets** of $\boldsymbol{q} \pmod{n}$.

EXAMPLE 5.11.2

The cyclotomic cosets of 2 modulo 15 ($q = 2$, $n = 15$) are:

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1, 2, 4, 8\} = C_2 = C_4 = C_8 \\ C_3 &= \{3, 6, 12, 9\} = C_6 = C_{12} = C_9 \\ C_5 &= \{5, 10\} = C_{10} \\ C_7 &= \{7, 14, 13, 11\} = C_{14} = C_{13} = C_{11} \end{aligned}$$

As the example suggests, if $j \in C_i$, then $C_j = C_i$.

Note:

$$\begin{aligned} m_{\beta^i}(x) &= (x - \beta^i)(x - \beta^{iq})(x - \beta^{iq^2}) \cdots (x - \beta^{iq^{t-1}}) \\ &= \prod_{j \in C_i} (x - \beta^i) \end{aligned}$$

is an irreducible factor of $x^n - 1$ over $GF(q)$ of degree $|C_i|$.

THEOREM 5.11.3

Suppose $\gcd(n, q) = 1$.

(i) The number of irreducible factors of $x^n - 1$ over $GF(q)$ is equal to the number of (distinct) cyclotomic cosets of $q \pmod n$.

(ii) The number of irreducible factors of degree d is equal to the number of (distinct) cyclotomic cosets of $q \pmod n$ of size d.

Alternatively,

THEOREM 5.11.4

Suppose $\gcd(n, q) = 1$. *Let $\beta \in GF(q^m)$ have order n, where m is the smallest positive integer such that $q^m \equiv 1 \pmod n$. Then, the irreducible factors of $x^n - 1$ over $GF(q)$ are*

$$\{m_{\beta^i}(x) : 0 \leqslant i \leqslant n - 1\}$$

where

$$m_{\beta^i}(x) = \prod_{j \in C_i} (x - \beta^j)$$

Note: If $j \in C_i$, then $m_{\beta^i}(x) = m_{\beta^j}(x)$.

EXAMPLE 5.11.5

Factor $x^{15} - 1$ over $GF(2)$ ($q = 2$, $n = 15$).

Solution. We know from the cyclotomic cosets of $2 \pmod{15}$ that $x^{15} - 1$ has 5 irreducible factors over $GF(2)$.

- 1 of degree 1
- 1 of degree 2
- 3 of degree 4

Let's find them. The smallest m such that $2^m \equiv 1 \pmod{15}$ is $m = 4$. We need an element β of order 15 in $GF(2^4)$; we can take $\beta = \alpha$ where $\alpha = x$ is a generator of $GF(2^4)^*$, where $GF(2^4) = \mathbb{Z}_2[x]/(x^4 + x + 1)$. In Example 5.10.9, we listed the powers of $\alpha = x$, and we computed

$$m_{\alpha^6}(x) = 1 + x + x^2 + x^3 + x^4$$

Similarly (left as an exercise), we can compute:

$$\begin{aligned} m_{\alpha^0}(x) &= 1 + x \\ m_{\alpha^1}(x) &= 1 + x + x^4 \\ m_{\alpha^3}(x) &= 1 + x + x^2 + x^3 + x^4 \\ m_{\alpha^5}(x) &= (x - \alpha^5)(x - \alpha^{10}) = 1 + x + x^2 \\ m_{\alpha^7}(x) &= 1 + x^3 + x^4 \end{aligned}$$

Thus,

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

EXAMPLE 5.11.6

Determine the number of cyclic subspaces of $V_{90}(\mathbb{Z}_3)$.
Solution. First, observe that $x^{90} - 1 = (x^{10} - 1)^9$. To determine the factorization pattern of $x^{10} - 1$ over $\mathbb{Z}_3$, we need to find the cyclotomic cosets of $q = 3 \pmod{n = 10}$:

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1, 3, 9, 7\} \\ C_2 &= \{2, 6, 8, 4\} \\ C_5 &= \{5\} \end{aligned}$$

Therefore, $x^{90} - 1 = (f_0 f_1 f_2 f_5)^9$ where $\deg(f_0) = 1$, $\deg(f_1) = 4$, $\deg(f_2) = 4$, and $\deg(f_5) = 1$ and f_0, f_1, f_2, f_5 are irreducible over $\mathbb{Z}_3[x]$. Thus, the number of cyclic subspaces of $V_{90}(\mathbb{Z}_3)$ is

$$10 \times 10 \times 10 \times 10 = 10000$$

Note:

$$\begin{aligned} f_0(x) &= m_{\beta^0}(x) \\ f_1(x) &= m_{\beta^1}(x) \\ f_2(x) &= m_{\beta^2}(x) \\ f_5(x) &= m_{\beta^5}(x) \end{aligned}$$

where β is an element of order 10 in $GF(3^4)$ since $3^4 \equiv 1 \pmod{10}$.

Chapter 6

BCH Codes and Bounds for Cyclic Codes

2020-03-25

6.1 Introduction

BCH codes are cyclic codes which are constructed in such a way that a lower bound on their distance is known.

6.2 BCH Codes and the BCH Bound

Setup

- Assume $\gcd(n, q) = 1$.
- Let m be the smallest positive integer such that $q^m \equiv 1 \pmod{n}$.
- Let α be a generator of $GF(q^m)^*$, and let $\beta = \alpha^{(q^m-1)/n}$, so $\text{ord}(\beta) = n$.
- Let $m_{\beta^i}(x)$ denote the minimal polynomial of β^i over $GF(q)$ for a fixed integer $i \in [0, n-1]$.
- We will let $m_{\beta^i}(x) = m_{\beta^{i \pmod n}}(x)$ for $i \geqslant n$ since $\beta^i = \beta^{i \pmod n}$.

DEFINITION 6.2.1: BCH code, Designed distance

A **BCH code** C over $GF(q)$ of block length n and **designed distance** δ is a cyclic code generated by

$$g(x) = \text{lcm}\{m_{\beta^i}(x) : a \leqslant i \leqslant a + \delta - 2\}$$

for some $a \in \mathbb{Z}$.

Notes:

(i) $\text{lcm}(3, 3, 5, 7, 7, 7, 11, 11) = 3 \times 5 \times 7 \times 11$.

(ii) $m_{\beta^i}(x) \mid (x^n - 1)$ for each i, $a \leqslant i \leqslant a + \delta - 2$, it follows that $g(x) \mid (x^n - 1)$. Also, $g(x)$ is monic. Hence, $g(x)$ is indeed the generator polynomial for a cyclic code of length n over $GF(q)$.

(iii) The $\delta - 1$ consecutive powers of β: $\beta^a, \beta^{a+1}, \ldots, \beta^{a+\delta-2}$ are roots of $g(x)$.

(iv) **BCH bound**: $d(C) \geqslant \delta$.

EXAMPLE 6.2.2: Constructing a BCH Code

Let $q = 3$, $n = 13$. Then, $m = 3$ since $3^3 \equiv 1 \pmod{13}$. Consider $GF(3^3) = \mathbb{Z}_3[x]/(x^3 + 2x^2 + 1)$. Then, $\alpha = x$ is a generator of $GF(3^3)^*$ as the following table shows.

$$\begin{array}{ccc}
\alpha^0 = 1 & \alpha^9 = 2 + 2\alpha + \alpha^2 & \alpha^{18} = 1 + \alpha \\
\alpha^1 = \alpha & \alpha^{10} = 1 + 2\alpha + 2\alpha^2 & \alpha^{19} = \alpha + \alpha^2 \\
\alpha^2 = \alpha^2 & \alpha^{11} = 2 + \alpha & \alpha^{20} = 2 + 2\alpha^2 \\
\alpha^3 = 2 + \alpha^2 & \alpha^{12} = 2\alpha + \alpha^2 & \alpha^{21} = 1 + 2\alpha + 2\alpha^2 \\
\alpha^4 = 2 + 2\alpha + \alpha^2 & \alpha^{13} = 2 & \alpha^{22} = 1 + \alpha + \alpha^2 \\
\alpha^5 = 2 + 2\alpha & \alpha^{14} = 2\alpha & \alpha^{23} = 2 + \alpha + 2\alpha^2 \\
\alpha^6 = 2\alpha + 2\alpha^2 & \alpha^{15} = 2\alpha^2 & \alpha^{24} = 1 + 2\alpha \\
\alpha^7 = 1 + \alpha^2 & \alpha^{16} = 1 + 2\alpha^2 & \alpha^{25} = \alpha + 2\alpha^2 \\
\alpha^8 = 2 + \alpha + \alpha^2 & \alpha^{17} = 1 + \alpha + 2\alpha^2 & \alpha^{26} = 1
\end{array}$$

Also, $\beta = \alpha^2$ is an element of order 13.
Compute the cyclotomic cosets of $q = 3 \mod 13 = n$:

$$\begin{aligned}
C_0 &= \{0\} \\
C_1 &= \{1, 3, 9\} \\
C_2 &= \{2, 6, 5\} \\
C_4 &= \{4, 12, 10\} \\
C_7 &= \{7, 8, 11\}
\end{aligned}$$

The corresponding minimal polynomials are:

$$\begin{aligned}
m_{\beta^0}(x) &= x + 2 \\
m_{\beta^1}(x) &= x^3 + 2x^2 + 2x + 2 \\
m_{\beta^2}(x) &= x^3 + 2x + 2 \\
m_{\beta^4}(x) &= x^3 + x^2 + x + 2 \\
m_{\beta^7}(x) &= x^3 + 2x + 1
\end{aligned}$$

Arithmetic of $m_{\beta^2}(x)$

$$\begin{aligned}
m_{\beta^2}(x) &= (x - \beta^2)(x - \beta^6)(x - \beta^5) \\
&= (x - \alpha^4)(x - \alpha^{12})(x - \alpha^{10}) \\
&= \left[x^2 - (\alpha^4 + \alpha^{12})x + \alpha^{16}\right](x - \alpha^{10}) \\
&= (x^2 + \alpha^{10}x + \alpha^{16})(x + \alpha^{23}) \\
&= x^3 + (\alpha^{10} + \alpha^{23})x^2 + (\alpha^{16} + \alpha^{33})x + \alpha^{39} \\
&= x^3 + 2x + 2
\end{aligned}$$

Let

$$g(x) = m_{\beta^0}(x)m_{\beta^1}(x)m_{\beta^2}(x) = 2 + 2x + x^4 + 2x^5 + x^6 + x^7$$

The roots of $g(x)$ are: $\beta^0, \beta^1, \beta^3, \beta^9, \beta^2, \beta^6, \beta^5$.
Since $\beta^0, \beta^1, \beta^2, \beta^3$ are among these roots, $\delta = 5 \implies d \geqslant 5$.
Thus, $g(x)$ generates a $(13, 6)$-BCH code over $GF(3)$ of distance at least 5.

EXERCISE 6.2.3

Show that

$$g(x) = m_{\beta^0}(x)m_{\beta^4}(x)m_{\beta^7}(x)$$

generates a $(13, 6)$-BCH code over $GF(3)$ of distance at least 5.

EXAMPLE 6.2.4

Does there exist a block code with parameters $q = 2$, $n = 128$, $M = 2^{64}$, and $d \geqslant 22$?
The corresponding *sphere-packing problem* is:
Can we place 2^{64} spheres of radius $\geqslant 10$ in $V_{128}(\mathbb{Z}_2)$ so that no two spheres intersect?
Solution. Yes! We will describe an **extended BCH code** with these parameters.
Let $q = 2$ and $n = 127$. The cyclotomic cosets of $2 \pmod{127}$ are:

$$\begin{array}{cc} C_0 = \{0\} & C_{11} = \{11, 22, 44, 88, 49, 98, 69\} \\ C_1 = \{1, 2, 4, 8, 16, 32, 64\} & C_{13} = \{13, 26, 52, 104, 81, 35, 70\} \\ C_3 = \{3, 6, 12, 24, 48, 96, 65\} & C_{15} = \{15, 30, 60, 120, 113, 99, 71\} \\ C_5 = \{5, 10, 20, 40, 80, 33, 66\} & C_{19} = \{19, 38, 76, 25, 50, 100, 73\} \\ C_7 = \{7, 14, 28, 56, 112, 97, 67\} & \vdots \\ C_9 = \{9, 18, 36, 72, 17, 34, 68\} & \end{array}$$

We have $m = 7$. Let β be an element of order 127 in $GF(2^7)^*$. Then,

$$g(x) = m_{\beta^1}(x)m_{\beta^3}(x)m_{\beta^5}(x)m_{\beta^7}(x)m_{\beta^9}(x)m_{\beta^{11}}(x)m_{\beta^{13}}(x)m_{\beta^{15}}(x)m_{\beta^{19}}(x)$$

is a degree-63 divisor of $x^{127} - 1$ over $GF(2)$.
Moreover, the roots of $g(x)$ include the follow 20 consecutive powers of β: $1, 2, \ldots, 20$.
Thus, $g(x)$ generates a binary $(127, 64)$-BCH code C with distance $\geqslant 21$.
Finally, the extended code of C (i.e., the code obtained by adding a parity bit to each codeword in C–see A2Q5) is a binary $(128, 64)$-code with distance $\geqslant 22$.
Note: The rate of the code is $1/2$.

DEFINITION 6.2.5: Vandermonde matrix

A **Vandermonde matrix** over a field F is an $n \times n$ matrix of the form

$$A(x_1, x_2, x_3, \ldots, x_n) = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}$$

where $x_i \in F$.

THEOREM 6.2.6

$\det(A) \neq 0$ *if and only if* x_i *are pairwise distinct.*

THEOREM 6.2.7: BCH Bound

Let C be an (n, k)-BCH code over $GF(q)$ with designed distance δ. Then, $d(C) \geqslant \delta$.

Proof of Theorem 6.2.7

Let $g(x)$ be the generator polynomial for C. Suppose that $\beta, \beta^2, \ldots, \beta^{\delta-1}$ are the roots of $g(x)$, where $\beta \in GF(q^m)$ is an element of order n. For simplicity, we have chosen $a = 1$. Hence,

$$g(x) = \text{lcm}\{m_{\beta^i}(x) : 1 \leqslant i \leqslant \delta - 1\}$$

Now, let $\boldsymbol{r} \in V_n(GF(q))$. Then,

$$\begin{aligned} \boldsymbol{r} \in C &\iff g(x) \mid r(x) \\ &\iff m_{\beta^i}(x) \mid r(x) \qquad \forall i \in [1, \delta - 1] \\ &\iff r(\beta^i) = 0 \qquad \forall i \in [1, \delta - 1] \end{aligned}$$

Let

$$H_1 = \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{n-1} \\ 1 & \beta^2 & (\beta^2)^2 & \cdots & (\beta^2)^{n-1} \\ \vdots & & & & \\ 1 & \beta^{\delta-1} & (\beta^{\delta-1})^2 & \cdots & (\beta^{\delta-1})^{n-1} \end{bmatrix}_{(\delta-1)\times n}$$

Now, $\boldsymbol{r} \in C \iff H_1 \boldsymbol{r}^\top = \boldsymbol{0}$. Furthermore, no $t = \delta - 1$ columns of H_1 are linearly dependent over $GF(q^m)$ since

$$\det \begin{bmatrix} \beta^{i_1} & \beta^{i_2} & \cdots & \beta^{i_t} \\ (\beta^2)^{i_1} & (\beta^2)^{i_2} & \cdots & (\beta^2)^{i_t} \\ \vdots & & & \\ (\beta^{\delta-1})^{i_1} & (\beta^{\delta-1})^{i_2} & \cdots & (\beta^{\delta-1})^{i_t} \end{bmatrix}_{t\times t} = \prod_{j=1}^{t} \beta^{i_j} \det[\underbrace{A(\beta^{i_1}, \ldots, \beta^{i_t})}_{\text{Vandermonde Matrix}}] \neq 0$$

since $\beta^{i_1}, \ldots, \beta^{i_t}$ are distinct.

Since $GF(q) \subseteq GF(q^m)$, we also have that no $\delta - 1$ columns of H_1 are linearly dependent over $GF(q)$. Now, if $\boldsymbol{c} \in C$, $\boldsymbol{c} \neq \boldsymbol{0}$, $w(\boldsymbol{c}) < \delta$, then $H_1 \boldsymbol{c}^\top = \boldsymbol{0}$ gives 0 as a non-trivial linear combination of $\delta - 1$ (or fewer) columns of H_1, contradicting the fact what we just proved. Hence, every non-zero codeword in C has weight $\geqslant \delta$. Thus, $d(C) \geqslant \delta$.

2020-03-30

6.3 Decoding BCH Codes

Over the years, many efficient algorithms have been designed for decoding BCH codes. One such algorithm is described in pages 215–219 of the course textbook. This algorithm is rather complicated. Instead of studying this algorithm, I will present a decoding algorithm for one specific BCH code, called C_{15}. The decoding algorithm for C_{15} captures the essential idea of a more general decoding algorithm for all BCH codes.

DEFINITION 6.3.1: C_{15}

Let $q = 2$, $n = 15$, $m = 4$. Let $GF(2^4) = \mathbb{Z}_2[x]/(x^4 + x + 1)$. Then, $\alpha = x$ is a generator of $GF(2^4)^*$ and $\beta = \alpha$ is an element of order 15.
Let

$$\begin{aligned} g(x) &= m_\beta(x) m_{\beta^3}(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) \\ &= 1 + x^4 + x^6 + x^7 + x^8 \end{aligned}$$

The roots of $g(x)$ include $\beta, \beta^2, \beta^3, \beta^4$. So, $g(x)$ generates a $(15, 7)$-BCH code over $GF(2)$ with $\delta = 5$, so $d \geqslant 5$. In fact, $d = 5$ since $g(x)$ has weight 5.
This BCH code is called $\boldsymbol{C_{15} : (15, 7, 5)}$**-binary code**.
Note: C_{15} is a 2-error correcting code.

Computing Syndromes

Let's first find a PCM for C_{15}. Let $\boldsymbol{r} \in V_{15}(\mathbb{Z}_2)$. Then

$$\begin{aligned} \boldsymbol{r} \in C_{15} &\iff g(x) \mid r(x) \\ &\iff m_\beta(x) \mid r(x) \text{ and } m_{\beta^3}(x) \mid r(x) \\ &\iff r(\beta) = 0 \text{ and } r(\beta^3) = 0. \end{aligned}$$

So, a PCM for C_{15} is

$$H = \begin{bmatrix} \beta^0 & \beta^1 & \beta^2 & \beta^3 & \cdots & \beta^{14} \\ (\beta^3)^0 & (\beta^3)^1 & (\beta^3)^2 & (\beta^3)^3 & \cdots & (\beta^3)^{15} \end{bmatrix}_{8 \times 15}$$

Note: H is a 2×15 matrix over $GF(2^4)$, and an 8×15 matrix over $GF(2)$.

Syndromes

The syndrome of $\boldsymbol{r}$ is

$$H\boldsymbol{r}^\top = \begin{bmatrix} r(\beta) \\ r(\beta^3) \end{bmatrix} = \begin{bmatrix} s_1 \\ s_3 \end{bmatrix}$$

(So, we don't need H to compute syndromes)

Recall: C_{15} is a $(15, 7, 5)$-BCH code over $GF(2)$. The *syndrome* of $\boldsymbol{r}$ consists of $s_1 = r(\beta)$ and $s_3 = r(\beta^3)$. We have $s_1, s_3 \in GF(2^4)$.

Decoding strategy

If there is an error vector $\boldsymbol{e}$ of weight at most 2, that has syndrome (s_1, s_3), then we decode $\boldsymbol{r}$ to $\boldsymbol{r} - \boldsymbol{e}$. Otherwise, we reject $\boldsymbol{r}$.

Decoding Algorithm for C15 [With Justification]

- Received word is $\boldsymbol{r} \in V_{15}(GF(2))$.
- Compute $s_1 = r(\beta)$ and $s_3 = r(\beta^3)$.
- If $s_1 = 0$ and $s_3 = 0$, then accept $\boldsymbol{r}$; STOP.
- Suppose $e(x) = x^i$; i.e., exactly one error has occurred in the i^{th} position $i \in [0, 14]$. Then, $s_1 = r(\beta) = c(\beta) + e(\beta) = e(\beta) = \beta^i$, and $s_3 = r(\beta^3) = e(\beta^3) = \beta^{3i}$. Hence, $s_3 = s_1^3$. If $s_1^3 = s_3$, then correct $\boldsymbol{r}$ in position i where $s_1 = \beta^i$; STOP.

- If $s_1 = 0$ (and $s_3 \neq 0$), then reject $\boldsymbol{r}$; STOP. Since $r(\beta^3) = e(\beta^3) \neq 0$, we have $e(x) \neq 0$. If $s_1 = r(\beta) = 0$, then $e(\beta) = 0$, so $m_\beta(x) \mid e(x)$, so $w(\boldsymbol{e}) \geqslant 3$ since the BCH code generated by $m_\beta(x)$ has $\delta \geqslant 3$.
- If exactly two errors have occurred, say in positions i and j with $i \neq j$ and $i, j \in [0, 14]$, then $e(x) = x^i + x^j$. Thus, $s_1 = r(\beta) = e(\beta) = \beta^i + \beta^j$ and

$$\begin{aligned} s_3 = r(\beta^3) &= e(\beta^3) \\ &= \beta^{3i} + \beta^{3j} \\ &= (\beta^i + \beta^j)(\beta^{2i} + \beta^{i+j} + \beta^{2j}) \\ &= (\beta^i + \beta^j)((\beta^i + \beta^j)^2 + \beta^{i+j}) \\ &= s_1(s_1^2 + \beta^{i+j}) \end{aligned}$$

  therefore, ${}^{s_3}/_{s_1} + s_1^2 = \beta^{i+j}$. Hence, β^i and β^j are the roots of the polynomial $z^2 + (\beta^i + \beta^j)z + \beta^{i+j} = z^2 + s_1 z + \left(\frac{s_3}{s_1} + s_1^2\right) = 0$. Form the **error locator polynomial** $\sigma(z) = z^2 + s_1 z + \left(\frac{s_3}{s_1} + s_1^2\right)$, and find its roots, if any, in $GF(2^4)$. If there are two roots, β^i and β^j, correct $\boldsymbol{r}$ in positions i and j; STOP.
- Reject $\boldsymbol{r}$.

Algorithm 5: Decoding Algorithm for C_{15}

```
Received word is r
s_1 ← r(β)
s_3 ← r(β^3)
if s_1 = 0 and s_3 = 0 then
    return r
if s_1^3 = s_3 then
    if s_1 = β^i then
        return (r_1, ..., r_15) where r_i ← r̄_i
if s_1 = 0 (and s_3 ≠ 0) then
    return
Form the error locator polynomial σ(z) = z^2 + s_1 z + (s_3/s_1 + s_1^2) and find its roots, if any, in GF(2^4)
if there are two (distinct) roots β^i and β^j then
    return corrected r in positions i and j
return
```

REMARK 6.3.2

We start position count from 0.

EXAMPLE 6.3.3: Decoding C_{15}

Decode $\boldsymbol{r} = (10001\ 00110\ 00000) \iff 1 + x^4 + x^7 + x^8$.

$$s_1 = r(\beta) = 1 + \beta^4 + \beta^7 + \beta^8 = \beta + \beta^{11} = \beta^6$$

$$s_3 = r(\beta^3) = 1 + \beta^{12} + \beta^6 + \beta^9 = \beta^3$$

$$s_1^3 = (\beta^6)^3 = \beta^{18} = \beta^3 = s_3,$$

so one error has occurred in position 6. So, correct $\boldsymbol{r}$ to

$$\boldsymbol{c} = (10001\ 01110\ 00000)$$

We can verify that $\boldsymbol{c} \in C_{15}$ by checking $g(x) \mid c(x)$ or check $c(\beta) = 0$ and $c(\beta^3) = 0$.

EXAMPLE 6.3.4: Decoding C_{15}

Decode $\boldsymbol{r} = (00111\ 01110\ 00000) \iff x^2 + x^3 + x^4 + x^6 + x^7 + x^8$.

$$s_1 = r(\beta) = \beta^2 + \beta^3 + \beta^4 + \beta^6 + \beta^7 + \beta^8 = \beta^{13}$$

$$s_3 = r(\beta^3) = \beta^6 + \beta^9 + \beta^{12} + \beta^3 + \beta^6 + \beta^9 = \beta^{10}$$

$$s_1^3 = \beta^{39} = \beta^9 \neq s_3$$

Error locator polynomial:

$$\sigma(z) = z^2 + s_1 z + \left(\frac{s_3}{s_1} + s_1^2\right) = z^2 + \beta^{13} z + (\beta^{12} + \beta^{11}) = z^2 + \beta^{13} z + 1$$

Let its roots be β^i and β^j. Then, $\beta^i \cdot \beta^j = 1 = \beta^0$. So, $i + j \equiv 0 \pmod{15}$. Hence, check if $\beta^i + \beta^j = \beta^{13}$ for

$$(i, j) \in \{(1, 14), (2, 13), (3, 12), (4, 11), (5, 10), (6, 9), (7, 8)\}$$

Discover that $\beta^4 + \beta^{11} = \beta^{13}$. So, correct $\boldsymbol{r}$ in positions 4 and 11:

$$\boldsymbol{c} = (00110\ 01110\ 01000)$$

More Generally

Suppose C is a binary (n, k)-BCH code with designed distance δ.

Suppose the generator polynomial of C is

$$g(x) = \text{lcm}\{m_{\beta^i}(x) : i \in [1, \delta - 1]\}$$

where β is an element of order n in $GF(2^m)$. Then, $d(C) \geqslant \delta$. Let $t = \lfloor \frac{\delta - 1}{2} \rfloor$.

Suppose $\boldsymbol{c} \in C$ is transmitted, $w(\boldsymbol{e}) \leqslant t$, and $\boldsymbol{r}$ is received.

Compute $s_i = r(\beta^i)$ for each $i \in [1, \delta - 1]$, and form the **syndrome polynomial**:

$$s(z) = s_1 + s_2 z + s_3 z^3 + \cdots + s_{\delta - 1} z^{\delta - 2}$$

Fact: From $s(z)$, the error locator polynomial can be efficiently computed. The roots of $\sigma(z)$ are β^{-j}, where j are the error positions.

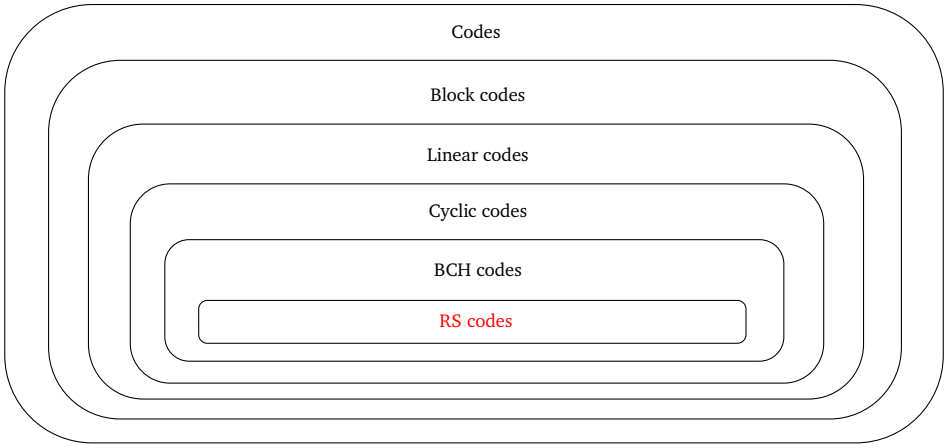
Chapter 7

Error Correction Techniques and Digital Audio Recording

2020-04-01

7.1 Reed-Solomon Codes

Invented by Irving Reed and Gustave Solomon in 1960.



DEFINITION 7.1.1: Reed-solomon code

A **Reed-Solomon (RS) code** is a BCH code of length n over $GF(q)$ where $n \mid (q-1)$.

REMARK 7.1.2

Since $q^1 \equiv 1 \mod n$, we have $m = 1$.

EXAMPLE 7.1.3

Let $q = 2^4$ and $GF(2^4) = \mathbb{Z}_2/(\alpha^4 + \alpha + 1)$. Recall that α is a generator of $GF(2^4)^\star$.
Let $\beta = \alpha^3$, then $\text{ord}(\beta) = 5$, (so $q = 16$, $n = 5$).

Let

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

Then, $g(x)$ generates a $(5,2)$-RS code C over $GF(2^4)$ with $\delta = 4$. In fact, $d(C) = 4$ since $g(x)$ is a codeword of weight 4.
A generator matrix for C is

$$G = \begin{bmatrix} \alpha^3 & \alpha^2 & \alpha^{11} & 1 & 0 \\ 0 & \alpha^3 & \alpha^2 & \alpha^{11} & 1 \end{bmatrix}_{2\times 5}$$

Consider the code C' obtained from C by replacing each symbol in codewords of C by their binary vector representation. For example,

$$\text{e.g.,} \quad (\alpha^3, \alpha^2, \alpha^{11}, 1, 0) \longleftrightarrow (0001\ 0010\ 0111\ 1000\ 0000)$$

It is not difficult to see that C' is closed under vector addition and scalar multiplication over $GF(2)$. Thus, C' is a $(20,8)$-binary code.

DEFINITION 7.1.4: RS code C of length n over $GF(q)$ with designed distance δ

Suppose $n \mid (q-1)$, and let $\beta \in GF(q)$ be an element of order n. Then, $m_{\beta^i}(x) = x - \beta^i$ for all i.
An **RS code C of length n over $GF(q)$ with designed distance δ** is a cyclic code over $GF(q)$ with generator polynomial

$$g(x) = (x-\beta^a)(x-\beta^{a+1})(x-\beta^{a+2})\cdots(x-\beta^{a+\delta-2}) \quad \text{for some } a.$$

Since $\deg(g) = \delta - 1$, we have $w(g) \leqslant \delta$, so $d(C) \leqslant \delta$. By the BCH bound, $d(C) \geqslant \delta$, hence $d(C) = \delta$.

Since $\dim(C) = k = n - \deg(g) = n - \delta + 1$, we have $k = n - d + 1$, so $d = n - k + 1$. Recall that $d \leqslant n - k + 1$ for any (n,k,d)-code. Thus, *RS codes are optimal* in the sense that, for any fixed n, k, q, they achieve maximum distance among all (n,k,d)-codes over $GF(q)$.

RS Codes Have Good (Cyclic) Burst Error Correcting Capability

- Let C be a RS code of length n over $GF(2^r)$ and designed distance δ. Consider $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in C$, and notice that $c_i \in GF(2^r)$. Let $e = \lfloor \frac{d-1}{2} \rfloor = \lfloor \frac{n-k}{2} \rfloor$.
- By writing each $\boldsymbol{c}_i$ as a binary vector of length r, we can view $\boldsymbol{c}$ as a binary vector of length nr bits.
- Now, if $\boldsymbol{c}$ is transmitted and if a cyclic burst error of length $\leqslant 1 + (e-1)r$ bits is introduced, then at most e symbols of $\boldsymbol{c}$ are received incorrectly. Thus, the received word can be decoded correctly.

THEOREM 7.1.5

Let C be an (n,k)-RS code over $GF(2^r)$. Then C', the code obtained by replacing each symbol in the codewords of C by the r-bit binary representations, is a binary (nr, kr)-code with cyclic burst error correcting capability $t = 1 + \left(\left\lfloor \frac{n-k}{2} \right\rfloor - 1 \right) r$.

EXAMPLE 7.1.6

Consider $GF(2^8) = \mathbb{Z}_2[\alpha]/(\alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1)$. Then $\beta = \alpha$ has order $n = 255$ (so $q = 256$, $n = 255$). Let

$$g(x) = \prod_{i=1}^{24} (x - \beta^i)$$

Then $g(x)$ is the generator polynomial for a $(255, 231, 25)$-RS code C with error correcting capability $e = 12$.

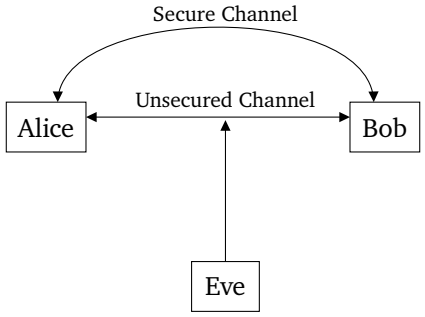
- The related code C' is a $(2040, 1848)$-binary code with cyclic burst error correcting capability $t = 89$.
- The code C, and others derived from it, have widely been used in practice, including in CDs, DVDs, and QR codes.

Chapter 8

Code-Based Cryptography

8.1 Public-Key Encryption

- **Goal**: Confidentiality, when communicating over an insecure channel.
- **Main feature**: The two communicating parties do not share any secrets. They only share *public information* that has been *authenticated*.



8.2 Basic RSA Encryption Scheme

Key Generation

Alice does the following:

1. Randomly select two large prime numbers, p and q.
2. Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$.
3. Select arbitrary e, $1 < e < \phi(n)$ with $\gcd(e, \phi(n)) = 1$.
4. Compute $d = e^{-1} \mod \phi(n)$.
5. Alice's public key is (n, e), while her private key is d. The only known way to recover d from the public key is to first factor n to find p and q. So, the private key remains secret to Alice as long is n is large enough so that no one else can factor it.

Encryption

To encrypt a message for Alice, Bob does:

1. Obtain an authentic copy of Alice's public key (n, e).
2. Represent the message m as an integer $[0, n-1]$.
3. Compute $c = m^e \mod n$.
4. Send c to Alice over the unsecured channel.

Decryption

To decrypt c, Alice does:

1. Compute $m = c^d \mod n$.

8.3 The Threat of Quantum Computers

- The security of RSA is based on the hardness of factoring n.
- It has been known since 1994 that factoring n is easy on a *quantum computer*.
- *Elliptic-curve cryptography*, a widely used alternative to RSA can also be broken easily by quantum computers.
- We are still *very far away* from being able to build large-scale quantum computers.
- Nonetheless, it seems prudent to develop public-key encryption schemes that *resist attacks even by quantum computers*.

8.4 McEliece Public-Key Encryption Scheme (1978)

- **Security** is based on the fact that decoding a random (binary) linear code is *NP-hard*.
- **Idea**:
  - Select a code C for which an efficient decoding algorithm is known.
  - Disguise C to get "random looking" code $\hat{C}$.
  - $\hat{C}$ is your *public key*, while the "disguising factor" is your *private key*.
  - **Encryption**: Encode $\boldsymbol{m}$ to get $\hat{\boldsymbol{c}} \in \hat{C}$, add a random error $\boldsymbol{e}$ to $\hat{\boldsymbol{c}}$ to get $\hat{\boldsymbol{r}}$ ($\hat{\boldsymbol{r}} = \hat{\boldsymbol{c}} + \boldsymbol{e}$), and send $\hat{\boldsymbol{r}}$.
  - **Decryption**: Use the "disguising factor" to convert the decoding problem to one for C ($\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$), and then use the decoding algorithm for C to recover $\boldsymbol{e}$ and $\boldsymbol{m}$.

Key Generation

Alice does the following:

1. Select a $k \times n$ generator matrix G for a t-error correcting **binary Goppa code** C.
2. Select a random $k \times k$ binary invertible matrix S.
3. Select a random $n \times n$ permutation matrix P.
4. Compute $\hat{G} = SGP$ ($\hat{G}$ is a $k \times n$ matrix of rank k).
5. Alice's *public key* is $(\hat{G}, t)$, while her *private key* is (G, S, P).

Conjecture: $\hat{G}$ is indistinguishable from a random $k \times n$ binary matrix of rank k.

Encryption

To encrypt a message for Alice, Bob does:

1. Obtain an authentic copy of Alice's public key $(\hat{G}, t)$.
2. Represent the message as a binary vector $\boldsymbol{m}$ of length k.
3. Select a random binary vector $\boldsymbol{e} \in V_n(\mathbf{Z}_2)$ of weight t.
4. Compute $\hat{\boldsymbol{r}} = \boldsymbol{m}\hat{G} + \boldsymbol{e}$, and send $\hat{\boldsymbol{r}}$ to Alice.

Decryption

To decrypt $\hat{\boldsymbol{r}}$, Alice does the following:

1. Compute $\boldsymbol{r} = \hat{\boldsymbol{r}}P^{-1}$.

   [Note: $\boldsymbol{r} = \hat{\boldsymbol{r}}P^{-1} = \boldsymbol{m}\hat{G}P^{-1} + \boldsymbol{e}P^{-1} = (\boldsymbol{m}SGP)P^{-1} + \boldsymbol{e}P^{-1} = (\boldsymbol{m}S)G + \boldsymbol{e}P^{-1}$]
2. Use the decoding algorithm for C to recover $\boldsymbol{m}' = \boldsymbol{m}S$.
3. Compute $\boldsymbol{m} = \boldsymbol{m}'S^{-1}$.

Security is based on the hardness of decoding $\hat{C}$ (the code generated by $\hat{G}$).

8.5 Implementation Notes

- **Suggested parameters**: $n = 4096$, $k = 3496$, and $t = 50$.
- Encryption is very fast.
- Decryption is relatively fast.
- Appears to resist quantum attacks.

Using Other Codes

Proposals that replace the Goppa codes with RS codes, LDPC codes, convolutional codes, etc., have all been broken.

One secure alternative is to use "*quasi-cyclic MDPC codes*."