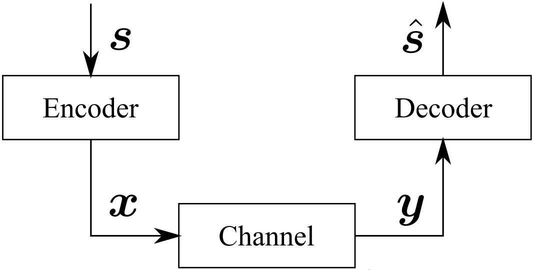
Section 3
Error Correcting Codes (ECC): Fundamentals

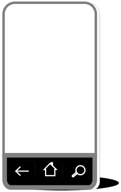
- Communication systems and channel models
- Definition and examples of ECCs
- Distance

For the contents relevant to distance, Lin & Xing's book, Chapter 2, should be helpful.

Communication Systems



redundant number

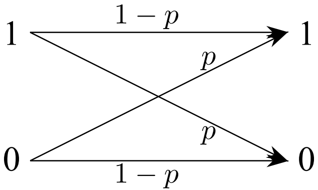


1 234567 890128 >

Abstract Channel Model: Binary Symmetric Channel (BSC)

Binary Symmetric Channel: a memoryless channel such that

$$\Pr\left(0 \text{ received} | 1 \text{ sent}\right) = \Pr\left(0 \text{ received} | 1 \text{ sent}\right) = p,$$
$$\Pr\left(1 \text{ received} | 1 \text{ sent}\right) = \Pr\left(0 \text{ received} | 0 \text{ sent}\right) = 1 - p.$$



p is called the transition (cross-over) probability.

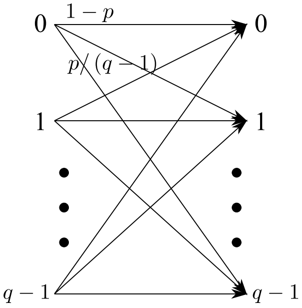
Memoryless channel: A channel that satisfies

$$\Pr\left(\boldsymbol{y} \text{ received} | \boldsymbol{x} \text{ sent}\right) = \prod_{i=1}^{n} \Pr\left(y_i \text{ received} | x_i \text{ sent}\right).$$

The Memoryless q-ary Symmetric Channel

Define an alphabet set $\mathbb{F}_q$.
Both channel input x_i and channel output y_i are from $\mathbb{F}_q$.

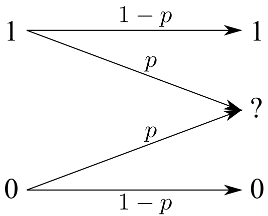
Crossover probability p:



$$\Pr(y_i|x_i) = \begin{cases} 1-p & \text{if } y_i = x_i \\ p/(q-1) & \text{if } y_i \neq x_i \end{cases}$$

The Memoryless Binary Erasure Channel (BEC)

Binary Erasure Channel:



- Internet traffic: a package got lost.
- Cloud storage: one copy of file messed up.

What is a Code?

Definition 3.1 (Code)

A code is a set $\mathcal{C}$ containing (row) vectors of elements from $\mathbb{F}_q$.
An (n, M) block code: $\mathcal{C} \subset \mathbb{F}_q^n$ and $|\mathcal{C}| = M$.

A codeword: a vector in $\mathcal{C}$.

Codeword length: n

Dimension: $k = \log_q M$

$|\mathcal{C}| = M$

Code size: M

Rate: $r = k/n$.

Example 1:

$q=2.\ n=4.\ M=3.\ k=\log_q M=\log_2 3.\ r=\frac{k}{n}=\frac{\log_2 3}{4}.$

$\mathbb{F}_2 = \{0,1\}$. $\mathcal{C} = \{0000,\ 1100,\ 1111\}$.
$n = 4.\ M = 3.\ k = \log_2 3 = 1.585.\ r = 0.3962.$

Example 2:

$q=3.\ n=5.\ M=3.\ k=\log_3 3=1.\ r=\frac{k}{n}=\frac{1}{5}.$

$\mathbb{F}_3 = \{0,1,2\}$. $\mathcal{C} = \{00000,\ 12121,\ 20202\}$.
$n = 5.\ M = 3.\ k = \log_3 3 = 1.\ r = 0.2.$

Triple Repetition Code

Encoding

$$1 \to 111 \qquad\qquad 0 \to 000$$

Decoding: majority voting

$$111,\ 110,\ 101,\ 011\ \to 1 \qquad\qquad 000,\ 001,\ 010,\ 100\ \to 0$$

Error probability computation:

$$\begin{aligned}
&P\left(\hat{s}=1|s=0\right) \\
&= P\left(111|0\right) + P\left(110|0\right) + P\left(101|0\right) + P\left(011|0\right) \\
&= p^3 + 3p^2\left(1-p\right) \\
&= 0.000298, \text{ when } p = 0.01.
\end{aligned}$$

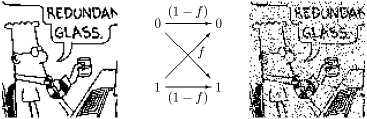
Much better than an uncoded system.

The price to pay: data rate $1/3$.

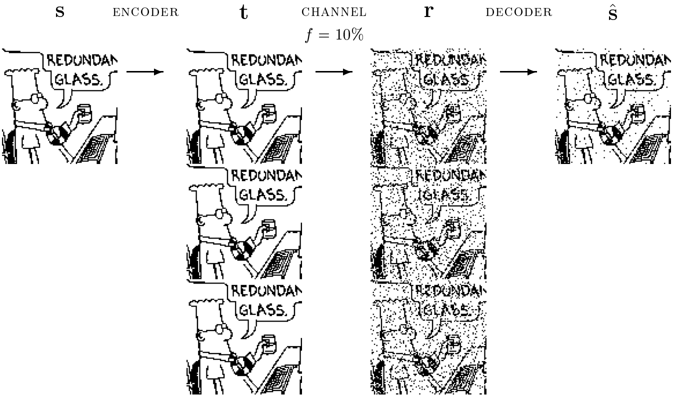
Coding theory: tradeoff between *error correction* and *data rate*.

Performance Comparison

Uncoded case (f=0.1)



Triple repetition code



From David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

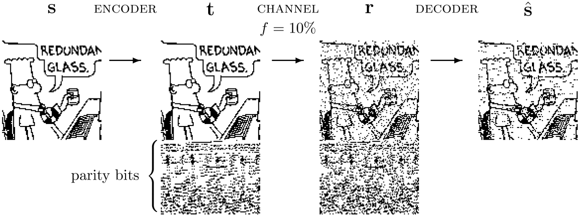
The 2nd example: $(7, 4)$ Hamming code

$r = \frac{k}{n}$ — dimension $k = \log_q M$. — codeword length.

Encoding: encode every 4 bit information into 7 bits. (Details are omitted.)

Code rate: $r = 4/7 \approx 0.57$.

Much higher rate but slightly larger P_e.



From David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

Another example - low-density parity-check code

Details are omitted here. Only simulation is presented

BSC with $p = 7.5\%$. LDPC $(20\,000,\ 10\,000)$ $r = 0.5$

iterative

From David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

Distance: Definition

Definition 3.2 (Distance)

A distance d on a set $\mathcal{X}$ is a function

$$d: \ \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

such that for all $x, y, z \in \mathcal{X}$, the following conditions hold:

- Positive definite:
$$d\left(x, y\right) \geq 0 \text{ where “=” holds iff } \boldsymbol{x} = \boldsymbol{y}.$$
- Symmetry:
$$d\left(x, y\right) = d\left(y, x\right).$$
- Triangle inequality:
$$d\left(x, z\right) \leq d\left(x, y\right) + d\left(y, z\right).$$

In this course, d is also translation invariant, that is,

$$d\left(x, y\right) = d\left(x + z, y + z\right).$$

Examples of Commonly Used Distances

Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ be two vectors of length n, for example,

$$\boldsymbol{x} = [9, 1, 0], \ \boldsymbol{y} = [6, 1, 4] \ \in \mathbb{R}^3$$

- ℓ_2-norm distance: Euclidean distance d_2

$$\begin{aligned} d_2(\boldsymbol{x}, \boldsymbol{y}) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\ &= \sqrt{3^2 + 0^2 + 4^4} = 5. \end{aligned}$$

- ℓ_1-norm distance: d_1

$$\begin{aligned} d_1(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^n |x_i - y_i| \\ &= 3 + 0 + 4 = 7. \end{aligned}$$

- Hamming distance: d_H

$$\begin{aligned} d_H(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^n \delta_{x_i \neq y_i} \\ &= 1 + 0 + 1 = 2, \end{aligned}$$

  where $\delta_{x_i \neq y_i} = 1$ if $x_i \neq y_i$ and $\delta_{x_i \neq y_i} = 0$ if $x_i = y_i$.

Hamming Distance

Definition 3.3 (Hamming Distance)

Hamming distance:
number of different elements.

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}^n$, the Hamming distance is given by

$$\begin{aligned} d_H(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^{n} \delta_{x_i \neq y_i} \\ &= |\{i : x_i \neq y_i\}| \end{aligned}.$$

Fact 3.4

Hamming distance is a well defined distance.

To prove this fact, the only non-trivial part is the triangle inequality.

Proof of the Triangle Inequality for Hamming Distance

1. scalar case: $d_H(x_i, z_i) \leq d_H(x_i, y_i) + d_H(y_i, z_i)$:
   If $x_i = z_i$, then the equality holds obviously.
   If $x_i \neq z_i$, LHS$= 1$. We have three cases:
   $$\left.\begin{array}{l} y_i = x_i \ \Rightarrow \ y_i \neq z_i \\ y_i = z_i \ \Rightarrow \ y_i \neq x_i \\ y_i \neq x_i \text{ and } y_i \neq z_i \end{array}\right\} \Rightarrow \text{RHS} \geq 1.$$
2. vector case:
   $$\begin{aligned} d_H(\boldsymbol{x}, \boldsymbol{z}) &= \sum_{i=1}^n d_H(x_i, z_i) \\ &\leq \sum_{i=1}^n \left(d_H(x_i, y_i) + d_H(y_i, z_i)\right) \\ &= d_H(\boldsymbol{x}, \boldsymbol{y}) + d_H(\boldsymbol{y}, \boldsymbol{z}). \end{aligned}$$

Hamming Distance: Properties

Fact 3.5

Hamming distance is translation invariant:

$$d_H(\boldsymbol{x}_1, \boldsymbol{x}_2) = d_H(\boldsymbol{x}_1 + \boldsymbol{y}, \boldsymbol{x}_2 + \boldsymbol{y}).$$

Definition 3.6 (Weight)

A weight of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ is defined as its Hamming distance from the zero vector:

$$\mathsf{wt}(\boldsymbol{x}) = d_H(\boldsymbol{x}, 0).$$

Example:

- $\boldsymbol{x} = [9,1,4]$, $\boldsymbol{y} = [0,1,4] \Rightarrow d_H(\boldsymbol{x}, \boldsymbol{y}) = 1.$
- $\boldsymbol{x} = [1,2,1,2,1]$, $\boldsymbol{y} = [2,0,2,0,2] \Rightarrow d_H(\boldsymbol{x}, \boldsymbol{y}) = 5.$
- $\boldsymbol{x} = [0,1,0,1] \Rightarrow \mathrm{wt}(\boldsymbol{x}) = 2.$

Decoding Techniques

Suppose that a codeword $\boldsymbol{c} \in \mathcal{C} \subset \mathbb{F}_q^n$ is transmitted and a word $\boldsymbol{y}$ is received. The decoding function is defined as the mapping

$$\begin{aligned} \mathcal{D}:\ \mathbb{F}_q^n &\to \mathcal{C} \\ \boldsymbol{y} &\mapsto \hat{\boldsymbol{c}} \in \mathcal{C}. \end{aligned}$$

Popular decoding strategies include

Maximum likelihood decoding:

$$\hat{\boldsymbol{c}}_{ML} = \mathcal{D}_{ML}(\boldsymbol{y}) = \arg\max_{\boldsymbol{c} \in \mathcal{C}} \Pr(\boldsymbol{y} \text{ received} | \boldsymbol{c} \text{ sent}).$$

Minimum distance decoding:

$$\hat{\boldsymbol{c}}_{MD} = \mathcal{D}_{MD}(\boldsymbol{y}) = \arg\min_{\boldsymbol{c} \in \mathcal{C}} d_H(\boldsymbol{y}, \boldsymbol{c}).$$

They are equivalent for many channels.

Equivalence Between ML and MD decoding

Theorem 3.7

Consider a memoryless binary symmetric channel (BSC) with cross-over probability $p < 1/2$. Then

(practical case)

$$\hat{\boldsymbol{c}}_{ML} = \hat{\boldsymbol{c}}_{MD}.$$

Proof: $\Pr(\underline{y} \mid \underline{c}) = \prod_{i=1}^{n} \Pr(y_i \mid c_i) = p^{d_H(\underline{y},\underline{c})} (1-p)^{n-d_H(\underline{y},\underline{c})}$

$= \left(\frac{p}{1-p}\right)^{d_H(\underline{y},\underline{c})} \cdot (1-p)^n$

$\because p < \frac{1}{2}$ $\therefore \frac{p}{1-p} < 1$

$\therefore$ when $d_H(\underline{y},\underline{c})$ is minimised, $\Pr(\underline{y} \mid \underline{c})$ is maximised.

$$\begin{aligned}
\Pr(\boldsymbol{y} \text{ received} | \boldsymbol{c} \text{ sent}) &= \prod_{i=1}^{n} \Pr(y_i \text{ received} | c_i \text{ sent}) \\
&= p^{d_H(\boldsymbol{y},\boldsymbol{c})} (1-p)^{n-d_H(\boldsymbol{y},\boldsymbol{c})} \\
&= (1-p)^n \left(\frac{p}{1-p}\right)^{d_H(\boldsymbol{y},\boldsymbol{c})}.
\end{aligned}$$

That $p < 1/2$ implies that $p/(1-p) < 1$. Hence, $\Pr(\boldsymbol{y} \text{ received}|\boldsymbol{c} \text{ sent})$ is a monotonically decreasing function of $d_H(\boldsymbol{y},\boldsymbol{c})$. The maximum $\Pr(\boldsymbol{y}|\boldsymbol{c})$ is achieved when $d_H(\boldsymbol{y},\boldsymbol{c})$ is minimized. $\diamondsuit$

Distance of a Code

Definition 3.8

The distance of a code $\mathcal{C}$ is defined as

$$d_H(\mathcal{C}) = \min_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C},\, \boldsymbol{x}_1 \neq \boldsymbol{x}_2} d_H(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

Notation: An (n, M, d)-code:

a code of codeword length n, size M, and distance d.

$q=2$

Example: Consider the binary code $n=5.\ M=3.\ d=2.$ $k=\log_q M=\log_2 3$

$$\mathcal{C} = \{00000, 00111, 11111\}.$$ $r=\frac{n}{k}=\frac{5}{\log_2 3}$

It is a binary $(5, 3, 2)$-code.

$q=3.$

Example: Consider the ternary code $n=6.\ M=3.\ d=3.$ $k=\log_q M=1.$

$$\mathcal{C} = \{000000, 000111, 111222\}.$$ $r=\frac{n}{k}=6.$

It is a ternary $(6, 3, 3)$-code.

Error Detection

Error detector: if the received word $\boldsymbol{y} \in \mathcal{C}$, let $\hat{\boldsymbol{c}} = \boldsymbol{y}$ and claim no error; if $\boldsymbol{y} \notin \mathcal{C}$, claim that errors happened.

error: received word is not codeword (i.e. not in the dict).

Theorem 3.9

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be an (n, M, d) code. The above error detector detects every pattern of up to $d-1$ many errors.

Proof:

1. Every pattern of $d-1$ many errors are detectable. Since at most $d-1$ many errors happened, $0 < d_H(\boldsymbol{c}, \boldsymbol{y}) < d := d(\mathcal{C})$ and $\boldsymbol{y} \notin \mathcal{C}$. The receiver will claim that errors happened.
2. Exists a pattern of d many errors that is not detectable. By the definition of the code distance, there exist $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}$ s.t. $d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) = d$. Suppose that $\boldsymbol{c}_1$ is the transmitted codeword and the channel errors happen to be $\boldsymbol{e} = \boldsymbol{c}_2 - \boldsymbol{c}_1$ (d errors happened). Then $\boldsymbol{y} = \boldsymbol{c}_2$ is received. As $\boldsymbol{c}_2 \in \mathcal{C}$, the detector claims that no error happened and set $\hat{\boldsymbol{c}} = \boldsymbol{c}_2$. $\Diamond$

Error Correction

Theorem 3.10

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be an (n, M, d) code. The minimum distance decoder can correct every pattern of up to $t := \lfloor (d-1)/2 \rfloor$ many errors.

$d=6 \Rightarrow t = \lfloor \frac{d-1}{2} \rfloor = 2$

Side mark:

c_1 c_2

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor = \begin{cases} \frac{d}{2} - 0.5 & \text{if } d \text{ is odd,} \\ \frac{d}{2} - 1 & \text{if } d \text{ is even.} \end{cases}$$

$d=7 \Rightarrow t = \lfloor \frac{d-1}{2} \rfloor = 3.$

c_1 c_2

and

$$\boxed{2t + 1 \leq d(\mathcal{C}) \leq 2t + 2}$$

Examples:

The previous ternary $(6, 3, 3)$ code is exactly 1-error-detecting.

Error Correction: Proof

Proof: Let $\mathcal{D}$ be the minimum distance decoder. Let $\boldsymbol{c}$ and $\boldsymbol{y}$ be the transmitted codeword and received word respectively. Let $\hat{\boldsymbol{c}} = \mathcal{D}_{MD}(\boldsymbol{y})$.

1. If $d_H(\boldsymbol{y}, \boldsymbol{c}) \le t = \lfloor (d-1)/2 \rfloor$, then $\hat{\boldsymbol{c}} = \boldsymbol{c}$.
   Suppose that $\hat{\boldsymbol{c}} \ne \boldsymbol{c}$. By the way the decoder $\mathcal{D}_{MD}$ is defined,
   $$d_H(\boldsymbol{y}, \hat{\boldsymbol{c}}) \le d_H(\boldsymbol{y}, \boldsymbol{c}) \le t.$$
   On the other hand, by the definition of the code distance,
   $$d \overset{\text{definition}}{\le} d_H(\boldsymbol{c}, \hat{\boldsymbol{c}}) \overset{\text{tri}}{\le} d_H(\boldsymbol{c}, \boldsymbol{y}) + d_H(\boldsymbol{y}, \hat{\boldsymbol{c}}) \le 2t \overset{\text{correction}}{\le} d-1,$$
   which is a contradiction.

   y t+1 c d c'

2. $\exists$ a pair $(\boldsymbol{c}, \boldsymbol{y}) \in \mathcal{C} \times \mathbb{F}_q^n$ s.t. $d_H(\boldsymbol{y}, \boldsymbol{c}) = t+1$ and it may happen that $\mathcal{D}_{MD}(\boldsymbol{y}) \ne \boldsymbol{c}$.

   $d_H(y, c) = t+1 \Rightarrow d_H(y, c') \ge d-(t+1)$
   $d_H(c, c') = d$

   By the definition of the code distance, $\exists\ \boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$ s.t. $d_H(\boldsymbol{c}, \boldsymbol{c}') = d$.

   $\because 2t+1 \le d \le 2t+2$

   W.l.o.g., assume the first d positions of $\boldsymbol{c}, \boldsymbol{c}'$ are different. Define $\boldsymbol{y}$ such that $y_i = c_i'$, $i = 1, 2, \cdots, t+1$ and $y_i = c_i$, $i = t+2, \cdots, n$. It is clear that $d_H(\boldsymbol{y}, \boldsymbol{c}) = t+1$ and

   $\therefore d_H(y, c') = t+1 = d_H(y, c)$

   $d_H(\boldsymbol{y}, \boldsymbol{c}') = d - (t+1) \le t+1 = d_H(\boldsymbol{y}, \boldsymbol{c})$. Hence, it may happen that $\hat{\boldsymbol{c}} = \mathcal{D}_{MD}(\boldsymbol{y}) \ne \boldsymbol{c}$. $\Diamond$

Section 4
Linear Codes

- Definition.
  - Generator matrices.
  - Parity-check matrices.
- Decoding.

Remark: Why linear codes? Low complexity in encoding, decoding, and distance computation.

For the contents relevant to distance, Lin & Xing's book, Chapter 2, should be helpful.

Linear Codes: Definition

Block codes: all codewords are of the same length $\mathcal{C} \subset \mathbb{F}_q^n$.

Definition 4.1 (Linear Codes)

A linear code is a code for which any linear combination of codewords is also a codeword. (all zeros is always a codeword).
That is, let $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C} \subset \mathbb{F}_q^n$. Then $\lambda \boldsymbol{u} + \mu \boldsymbol{v} \in \mathcal{C}$, $\forall \lambda, \mu \in \mathbb{F}_q$.

Example of linear codes:
$\mathcal{C} = \{0000,\ 0011,\ 1100,\ 1111\} \subset \mathbb{F}_2^4$.
$\mathcal{C} = \left\{ \boldsymbol{v} \in \mathbb{F}_2^4 :\ \text{wt}\,(\boldsymbol{v})\ \text{ is even.} \right\}$.
Example of nonlinear codes:
$\mathcal{C} = \{0000,\ 1100,\ 1111\}$.
$\mathcal{C} = \left\{ \boldsymbol{v} \in \mathbb{F}_3^4 :\ \text{wt}\,(\boldsymbol{v})\ \text{ is even.} \right\}$.

$$\begin{array}{r} 1111 \\ 1100 \\ \hline 2211 \end{array} \times$$

Basis

Definition 4.2 (Basis)

Let $\mathcal{B} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\} \subset \mathbb{F}^n$. It is a basis of a set $\mathcal{C} \subset \mathbb{F}^n$ if it satisfies the following conditions:

- Linear independence property:
  For all $\lambda_1, \cdots, \lambda_k \in \mathbb{F}$, if $\sum \lambda_i \boldsymbol{v}_i = \boldsymbol{0}$, then necessarily $\lambda_1 = \cdots = \lambda_k = 0$.
- The spanning property:
  For every $\boldsymbol{c} \in \mathcal{C}$, there exist $\lambda_1, \cdots, \lambda_k \in \mathbb{F}$ s.t. $\boldsymbol{c} = \sum_i \lambda_i \boldsymbol{v}_i$.

$\dim(\mathcal{C}) = k$: the # of vectors in a basis.
The basis $\mathcal{B}$ is not unique in general, but the dimension is.

Example: Let $\mathcal{C} = \{0000,\ 0011,\ 1100,\ 1111\}$.
$\mathcal{B}_1 = \{0011,\ 1100\}$ is a basis for $\mathcal{C}$.
$\mathcal{B}_2 = \{0011,\ 1111\}$ is also a basis for $\mathcal{C}$.
$\dim(\mathcal{C}) = 2$.

Construct a Basis

Definition 4.3 (Linear Span)

For any subset $\mathcal{V} \subset \mathbb{F}^n$, define $\langle \mathcal{V} \rangle$ as the linear span of $\mathcal{V}$:

$$\langle \mathcal{V} \rangle = \left\{ \sum \lambda_i \boldsymbol{v}_i : \ \lambda_i \in \mathbb{F}, \ \boldsymbol{v}_i \in \mathcal{V} \right\}.$$

Construct a basis for a linear code $\mathcal{C} \subset \mathbb{F}^n$:

1. From $\mathcal{C}$, take a nonzero vector, say $\boldsymbol{v}_1$.
2. Take a nonzero vector, say $\boldsymbol{v}_2$, from $\mathcal{C} - \langle \{\boldsymbol{v}_1\} \rangle$.
3. Take a nonzero vector, say $\boldsymbol{v}_3$, from $\mathcal{C} - \langle \{\boldsymbol{v}_1, \boldsymbol{v}_2\} \rangle$.
4. Continue this process, until $\mathcal{C} - \langle \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\} \rangle = \phi$.
5. Set $\mathcal{B} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$.

The Size of a Linear Code

Theorem 4.4

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a linear code and dim $(\mathcal{C}) = k$, then $|\mathcal{C}| = q^k$.

Proof:

1. $\dim(\mathcal{C}) = k \Rightarrow \exists$ a basis $\mathcal{B} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\}$ for $\mathcal{C}$.
2. $|\mathcal{C}| \le q^k$:
   Definition of the basis suggests $\mathcal{C} = \langle \mathcal{B} \rangle = \left\{ \sum_{i=1}^k \lambda_i \boldsymbol{v}_i : \ \lambda_i \in \mathbb{F}_q \right\}$.
   There are q^k many possible linear combinations. Hence, $|\mathcal{C}| \le q^k$ (repetition may exist).
3. $|\mathcal{C}| = q^k$:
   It suffices to show that there is no repetition.
   Let $\boldsymbol{\lambda}^{(1)} \neq \boldsymbol{\lambda}^{(2)}$. Let $\boldsymbol{x}^{(1)} = \sum_{i=1}^k \lambda_i^{(1)} \boldsymbol{v}_1$ and $\boldsymbol{x}^{(2)} = \sum_{i=1}^k \lambda_i^{(2)} \boldsymbol{v}_1$.
   Then $\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)} = \sum_{i=1}^k \left(\lambda_i^{(1)} - \lambda_i^{(2)} \right) \boldsymbol{v}_i \neq \boldsymbol{0}$ by linear independence of $\boldsymbol{v}_i$'s and the fact that $\boldsymbol{\lambda}^{(1)} \neq \boldsymbol{\lambda}^{(2)}$.
   There is no repetition in the set $\left\{ \sum_{i=1}^k \lambda_i \boldsymbol{v}_i : \ \lambda_i \in \mathbb{F}_q \right\}$. $\diamondsuit$

Generator Matrix

Definition 4.5 (Generator Matrix)

A generator matrix $\boldsymbol{G}$ for a linear code $\mathcal{C} \subset \mathbb{F}^n$ is a matrix whose rows form a basis for $\mathcal{C}$.

For a given $(\overset{\text{length}}{n}, \overset{\text{dimension}}{k})$ linear code $\mathcal{C} \subset \mathbb{F}^n$, it can be defined using its generator matrix $\boldsymbol{G} \in \mathbb{F}^{k \times n}$.

length

dim

The encoding function that maps information symbols to a codeword is given by

$$\begin{aligned} E: \quad & \mathbb{F}^k \to \mathcal{C} \subset \mathbb{F}^n \\ & \boldsymbol{s} \mapsto \boldsymbol{c} = \boldsymbol{s}\boldsymbol{G} \in \mathcal{C}. \end{aligned}$$

Remark:
Encoding of a linear code is efficient: vector-matrix product.
Encoding of a nonlinear code is via a look-up table and hence computationally less efficient.

Examples

Example 1: the $(3, 1)$ repetition code: $\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$.

Example 2: the $(7, 4)$ Hamming code.

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Example 3: the generator matrix is not unique.
$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ and $\boldsymbol{G}' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ generate the same code $\mathcal{C} = \{0000,\ 1010,\ 0101,\ 1111\} \subset \mathbb{F}_2^4$.

Dual Code

Definition 4.6 (Dual Code)

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a non-empty code. Its dual code $\mathcal{C}^\perp$ is defined as

$$\mathcal{C}^\perp = \left\{ \boldsymbol{v} \in \mathbb{F}_q^n : \ \boldsymbol{v}\boldsymbol{c}^T = \sum_i v_i c_i = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C} \right\}.$$

Lemma 4.7

For any non-empty code $\mathcal{C} \subset \mathbb{F}_q^n$ (linear or nonlinear), its dual code $\mathcal{C}^\perp$ is always linear.

Proof: Take arbitrary $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathcal{C}^\perp$. Then for all $\lambda_1, \lambda_2 \in \mathbb{F}_q$ and for all $\boldsymbol{c} \in \mathcal{C}$,

$$(\lambda_1 \boldsymbol{v}_1 + \lambda_2 \boldsymbol{v}_2)\, \boldsymbol{c}^T = \lambda_1 \boldsymbol{v}_1 \boldsymbol{c}^T + \lambda_2 \boldsymbol{v}_2 \boldsymbol{c}^T = 0,$$

which implies $\lambda_1 \boldsymbol{v}_1 + \lambda_2 \boldsymbol{v}_2 \in \mathcal{C}^\perp$. $\diamondsuit$

Parity Check Matrix

Definition 4.8 (Parity-Check Matrix)

A parity-check matrix $\boldsymbol{H}$ for a linear code $\mathcal{C} \subset \mathbb{F}_q^n$ is a generator matrix for the dual code $\mathcal{C}^{\perp}$.

For a code $\mathcal{C}\,[n, k]$, it holds that

- $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ and $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$.
- $\boxed{\boldsymbol{H} \cdot \boldsymbol{G}^T = \boldsymbol{0}.}$

Define a linear code via its parity-check matrix:

$$\mathcal{C} = \left\{ \boldsymbol{c} \in \mathbb{F}_q^n : \ \underline{\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{0}} \right\}.$$

Examples

$$H \cdot G^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- The $(3,1)$ repetition code:

$$\boldsymbol{G} = \overset{k \times n}{\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}} \text{ and } \boldsymbol{H} = \overset{(n-k) \times n}{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}.$$

- The $(7,4)$ Hamming code:

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

- A self-dual code is a code s.t. $\mathcal{C} = \mathcal{C}^{\perp}$,
  Example: $\mathcal{C} = \{0000,\ 1010,\ 0101,\ 1111\}$, where

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

  Self-dual codes do not exist for vector space $\mathbb{R}^n$ or $\mathbb{C}^n$.

Relation Between $\boldsymbol{G}$ and $\boldsymbol{H}$

Consider $\mathcal{C}[n,k] \subset \mathbb{F}_q^n$. Write $\boldsymbol{G}$ and $\boldsymbol{H}$ in systematic forms:

- Let $\boldsymbol{G} = [\boldsymbol{I}_k \ \boldsymbol{A}] \in \mathbb{F}_q^{k\times n}$, where $\boldsymbol{A} \in \mathbb{F}_q^{k\times(n-k)}$.
- Let $\boldsymbol{H} = [\boldsymbol{B} \ \boldsymbol{I}_{n-k}] \in \mathbb{F}_q^{(n-k)\times n}$, where $\boldsymbol{B} \in \mathbb{F}_q^{(n-k)\times k}$.

2×5 $G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$ A $2\times(5-2)$

3×5 $H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ B $(5-2)\times 2$

Lemma 4.9

It holds that $\boldsymbol{B} = -\boldsymbol{A}^T$.

$B = -A^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$

Proof:

$$\boldsymbol{H}\boldsymbol{G}^T = [\boldsymbol{B} \ \boldsymbol{I}_{n-k}] \begin{bmatrix} \boldsymbol{I}_k \\ \boldsymbol{A}^T \end{bmatrix} = \boldsymbol{B}\cdot\boldsymbol{I}_k + \boldsymbol{I}_{n-k}\cdot\boldsymbol{A}^T$$
$$= -\boldsymbol{A}^T + \boldsymbol{A}^T = \boldsymbol{0} \in \mathbb{F}_q^{(n-k)\times k}.$$

Systematic form:
Why? Easy to compute $\boldsymbol{H}$ from $\boldsymbol{G}$, and vice versa.
How? Gaussian-Jordan elimination.

Hamming Weight

\# of nonzero components.

Hamming Weight of $\boldsymbol{x}$: $\mathrm{wt}(\boldsymbol{x}) = |\{i : x_i \neq 0\}| = d(\boldsymbol{x}, \mathbf{0})$.

Theorem 4.10

For a linear code $\mathcal{C}$, $d_H(\mathcal{C}) = \min_{\boldsymbol{x} \in \mathcal{C} \backslash \{\mathbf{0}\}} wt(\boldsymbol{x})$.

Proof: $d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) = \mathrm{wt}(\boldsymbol{c}_1 - \boldsymbol{c}_2) = \mathrm{wt}(\boldsymbol{c}')$ for some $\boldsymbol{c}' \in \mathcal{C}$ (by the definition of linear codes).

Notation: $\mathcal{C}[n, k, d]$: n: codeword length; k: dimension; d: distance.

Distance and Parity Check Matrix

Theorem 4.11

Let $\mathcal{C}$ be a linear code defined by the parity-check matrix $\boldsymbol{H}$. Then that $d(\mathcal{C}) = d$ is equivalent to that

1. *Every $d-1$ columns of $\boldsymbol{H}$ are linearly independent.*
2. *There exist d linearly dependent columns.*

Two Confusing Concepts

spark: minimum # linearly dependent
rank: maximum # linearly independent

Given a matrix $\boldsymbol{H}$,

- spark: minimum number of linearly dependent columns
- column rank: maximum number of linearly independent columns.

Theorem 4.11 suggests that $\boxed{\operatorname{spark}(\boldsymbol{H}) = d(\mathcal{C})}$.

Example 4.12

- $\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$: $\operatorname{spark}(\boldsymbol{H}) = 3$ and $\text{column rank}(\boldsymbol{H}) = 2$.
- $\boldsymbol{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$ is an $n \times (n+1)$ matrix:
  $\operatorname{spark}(\boldsymbol{H}) = 2$ and $\text{column rank}(\boldsymbol{H}) = n$.

Application of Theorem 4.11: Binary Hamming Codes

Definition 4.13 (Binary Hamming Codes)

length dimension distance

The parity-check matrix of the binary Hamming code $\mathcal{H}\left[2^r-1, 2^r-1-r, 3\right]$ consists of all the nonzero binary vectors (columns) of length r. (Also denoted by $\mathcal{H}_r$.)

Example 4.14

The $\mathcal{H}_2\left[3,1,3\right]$ is given by

$$\boldsymbol{H} = \begin{array}{c} \left[\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right] \\ \begin{array}{ccc} 1 & 2 & 3 \end{array} \end{array},$$

and the $\mathcal{H}_3\left[7,4,3\right]$ is given by

$$\boldsymbol{H} = \begin{array}{c} \left[\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right] \\ \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \end{array}.$$

The Distance of Binary Hamming Codes

Corollary 4.15

The distance of a binary Hamming code is 3, i.e., $d(\mathcal{H}_r) = 3$.

Proof: We apply Theorem 4.11.

- That there is no zero column implies that the minimum number of linearly dependent columns is at least 2, i.e., $d(\mathcal{C}) = \mathrm{spark}(\boldsymbol{H}) \geq 2$.
- In the binary field, that every two columns are distinct implies that every two columns are linearly independent. Hence, $d(\mathcal{C}) = \mathrm{spark}(\boldsymbol{H}) \geq 3$.
- It is easy to see that there exist three columns that are linearly dependent (for example the first three columns). Therefore $d(\mathcal{C}) = 3$. $\diamond$

Corollary 4.16

Binary Hamming codes correct up to one error.

Theorem 4.11: Proof

$d(\mathcal{C}) = d \Leftrightarrow$ every $d-1$ columns of H are L.I.
exist d columns of H are L.D.

Proof: Let $\boldsymbol{h}_i$ be the i^{th} column of $\boldsymbol{H}$. $\forall \boldsymbol{c} \in \mathcal{C}$, let $i_1, \cdots, i_K$ be the locations where $c_i \neq 0$. By the definition of parity-check matrix,

$$\boldsymbol{0} = \sum_{i=1}^{n} c_i \boldsymbol{h}_i = \sum_{k=1}^{K} c_{i_k} \boldsymbol{h}_{i_k}.$$

$d(\mathcal{C}) = d \Rightarrow$Claim 2: $d(\mathcal{C}) = d$ implies that $\exists \boldsymbol{c} \in \mathcal{C}$ s.t. $\mathrm{wt}(\boldsymbol{c}) = d$. That is, $\sum_{k=1}^{d} c_{i_k} \boldsymbol{h}_{i_k} = \boldsymbol{0}$, or, $\boldsymbol{h}_{i_1}, \cdots, \boldsymbol{h}_{i_k}$ are linearly dependent.
$d(\mathcal{C}) = d \Rightarrow$Claim 1: Suppose not. $\exists \boldsymbol{h}_{i_1}, \cdots \boldsymbol{h}_{i_{d-1}}$ are linear dependent, i.e., $\sum_{k=1}^{d-1} x_{i_k} \boldsymbol{h}_{i_k} = \boldsymbol{0}$. Let $\boldsymbol{x} = \left[0 \cdots x_{i_1} \cdots x_{i_k} \cdots x_{i_{d-1}} \cdots 0\right]$. Then $\mathrm{wt}(\boldsymbol{x}) \leq d-1$ and $\boldsymbol{x} \in \mathcal{C}$. Hence $d(\mathcal{C}) \leq d-1$. A contradiction with $d(\mathcal{C}) = d$.

Claims 1&2$\Rightarrow d(\mathcal{C}) = d$: That every $d-1$ columns are linearly independent implies no nonzero codeword of weight $d-1$. That there exists d columns that are linearly dependent means the existence of a codeword of weight d. Hence $d(\mathcal{C}) = \min_{\boldsymbol{x} \in \mathcal{C} \backslash \{\boldsymbol{0}\}} \mathrm{wt}(\boldsymbol{x}) = d$. $\diamond$

Syndrome Vector

Let $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ be a parity-check matrix of a linear code $\mathcal{C}\,[n,k] \subset \mathbb{F}_q^n$. Suppose that the received word is given by $\boldsymbol{y} \in \mathbb{F}_q^n$.

Define the syndrome vector

$$\boldsymbol{s} := \boldsymbol{y}\boldsymbol{H}^T. = eH^T$$

It depends only on the error vector not the transmitted codeword.

In particular, let $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$ where $\boldsymbol{x} \in \mathcal{C}$ is the transmitted codeword and $\boldsymbol{e} \in \mathbb{F}_q^n$ is the error vector introduced by the channel. It holds that

$$\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T = (\boldsymbol{x}+\boldsymbol{e})\,\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T.$$

Syndrome Decoding

Syndrome decoding:

1. compute syndrome vector, $s = yH^T = eH^T$
2. find error vector with minimum weight (MD)
   $\hat{e} = \arg\min_e wt(e)$
3. decode:
   $\hat{c} = y - \hat{e}$.

MD decoding: Find $\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c} \in \mathcal{C}} d_H(\boldsymbol{c}, \boldsymbol{y})$.

Syndrome decoding:

1. For the received word $\boldsymbol{y}$, compute the syndrome vector: $\boldsymbol{s} := \boldsymbol{y}\boldsymbol{H}^T$.
2. Find the error vector $\boldsymbol{e}$ with the minimum weight: (MD decoding)

$$\hat{\boldsymbol{e}} = \arg\min_{\boldsymbol{e}} \mathrm{wt}(\boldsymbol{e}) \text{ s.t. } \boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T. \tag{1}$$

3. Decode $\boldsymbol{y}$ as $\hat{\boldsymbol{c}} = \boldsymbol{y} - \hat{\boldsymbol{e}}$.

Comments: In general, it is computationally challenging to solve (1). However, all practical codes have particular structures in the parity-check matrix so that the decoding problem can be solved efficiently.

Decoding of Binary Hamming Codes

Take $\mathcal{H}_3$ (Definition 4.13) as an example.

index = 1 ⇒ 1st bit error

$e = [1000000]$

Assume that $\boldsymbol{y} = [0111111]$. Find the MD decoded codeword $\hat{\boldsymbol{c}} \in \mathcal{C}$.

$$S^T = H \cdot y^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

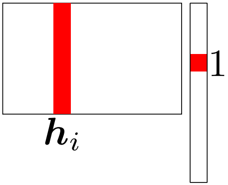
$\therefore \hat{c} = [1111111]$

Since $d(\mathcal{H}_3) = 3$, it corrects up to 1 error.

For any $\boldsymbol{e}$ s.t. $\mathrm{wt}(\boldsymbol{e}) = 1$, let $e_i \neq 0$ for some $i \in [n]$. Then

$$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T = e_i \boldsymbol{h}_i^T = \boldsymbol{h}_i^T.$$

$$\boldsymbol{s}^T = \boldsymbol{H}\boldsymbol{y}^T = \boldsymbol{H}\boldsymbol{e}^T =$$



$$= \boldsymbol{h}_i$$

In the example, $\boldsymbol{s} = [001]$, $\boldsymbol{e} = [1000000]$ and $\hat{\boldsymbol{c}} = [1111111]$.

Section 5
Coding Bounds

- Sphere packing (Hamming) bound
- Sphere covering (Gilbert-Varshamov) bound
- Singleton bound and MDS codes

The lectures will only cover sphere packing, sphere covering, singleton bounds and relevant contents. Reference: Lin & Xing's book, Chapter 5.

Coding Bounds: Motivation

Consider the Hamming code $\mathcal{H}_r$:

$r = 2$: $[3, 1, 3]$

$r = 3$: $[7, 4, 3]$

$r = 4$: $[15, 11, 3]$

Questions:

- Can we do better?
- What is the best that we can do?

larger d . more corrections

It is possible to construct linear codes with parameters

- $[7, 4, 4]$ over $\mathbb{F}_8$.
- $[15, 11, 5]$ over $\mathbb{F}_{16}$.

Hamming Bound

Theorem 5.1 (Hamming bound, sphere-packing bound)

For a code of length n and distance d, the number of codewords is upper bounded by

$$M \le q^n / \left(\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \right),$$

volume of balls

where $t := \left\lfloor \frac{d-1}{2} \right\rfloor$.

$\mathbb{F}_q^n = \{v\}$, $b = (x; t)$

$r=0$. center

positions parameters

\# points: $r=1 \Rightarrow |\{v : d_H(x,v) = 1\}| = \binom{n}{1}(q-1)$

$r=2 \Rightarrow |\{v : d_H(x,v) = 2\}| = \binom{n}{2}(q-1)^2$

$\vdots$

$r=t \Rightarrow |\{v : d_H(x,v) = t\}| = \binom{n}{t}(q-1)^t$

$\therefore$ volumn $= \sum$ points $= \sum_{i=0}^{t} \binom{n}{i}(q-1)^i$

$\therefore$ # balls = # of codewords

$M \le \dfrac{\text{available space}}{\text{ball volume}}$

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

Examples

Definition 5.2 (Perfect Codes)

A perfect code is a code that attains the Hamming bound.

- Binary Hamming code $\mathcal{H}_r\left[2^r-1, 2^r-1-r, 3\right]$ is a perfect code.
  $d=3 \Rightarrow t = \left\lfloor \frac{d-1}{2} \right\rfloor = 1$.
  Ball Volume: $\sum_{i=0}^{t} \binom{n}{i} (q-1)^i = 1 + (2^r - 1) = 2^r$.
  Hamming bound: $q^n / \sum_{i=0}^{t} \binom{n}{i} (q-1)^i = 2^{2^r-1}/2^r = 2^{2^r-r-1} = 2^k$.
- Perfect codes are rare (binary Hamming codes & Golay codes).

Hamming Bound: Proof (1)

Define a ball in $\mathbb{F}_q^n$ centered at $\boldsymbol{x} \in \mathbb{F}_q^n$ with radius t by

$$B(\boldsymbol{x}, t) = \left\{ \boldsymbol{y} \in \mathbb{F}_q^n : \ d(\boldsymbol{x}, \boldsymbol{y}) \le t \right\}.$$

Step one: the balls $B(\boldsymbol{c}, t)$, $\boldsymbol{c} \in \mathcal{C}$, are disjoint.
For all $\boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}$, it holds that $B(\boldsymbol{c}, t) \bigcap B(\boldsymbol{c}', t) = \phi$.
For a $\boldsymbol{y} \in B(\boldsymbol{c}, t)$, then $\boldsymbol{y} \notin B(\boldsymbol{c}', t)$ for all $\boldsymbol{c}' \neq \boldsymbol{c}$.

By triangle inequality: $d \le d_H(\boldsymbol{c}, \boldsymbol{c}') \le d_H(\boldsymbol{c}, \boldsymbol{y}) + d_H(\boldsymbol{y}, \boldsymbol{c}')$.
Then

$$\begin{aligned} d_H\left(\boldsymbol{y}, \boldsymbol{c}'\right) &\ge d - d_H(\boldsymbol{c}, \boldsymbol{y}) \ge d - t = d - \left\lfloor \frac{d-1}{2} \right\rfloor \\ &> \left\lfloor \frac{d-1}{2} \right\rfloor = t, \end{aligned}$$

which implies $\boldsymbol{y} \notin B(\boldsymbol{c}', t)$.

Hamming Bound: Proof (2)

Step two: Consider the union of these balls.

Clearly $\bigcup_{\boldsymbol{c}\in\mathcal{C}} B\left(\boldsymbol{c},t\right) \subset \mathbb{F}_q^n$. Hence,

$$\mathrm{Vol}\left(\bigcup_{\boldsymbol{c}\in\mathcal{C}} B\left(\boldsymbol{c},t\right)\right) = \sum_{\boldsymbol{c}\in\mathcal{C}} \mathrm{Vol}\left(B\left(\boldsymbol{c},t\right)\right) \le \mathrm{Vol}\left(\mathbb{F}_q^n\right) = q^n,$$

where the first equality holds because the balls do not overlap.

The volume of each ball is

$$\mathrm{Vol}\left(B\left(\boldsymbol{c},t\right)\right) = \sum_{i=0}^{t} \binom{n}{i} \left(q-1\right)^i .$$

Therefore

$$M\mathrm{Vol}\left(B\left(\boldsymbol{c},t\right)\right) \le q^n \quad \Rightarrow \quad M \le q^n / \sum_{i=0}^{t} \binom{n}{i} \left(q-1\right)^i .$$

$\diamond$

Gilbert-Varshamov Bound

Theorem 5.3 (Gilbert-Varshamov bound, sphere covering bound)

For given code length n and distance d, there exists a code such that

$$q^n/\mathrm{Vol}\left(d-1\right) \le M,$$

where $\mathrm{Vol}\left(d-1\right) := \sum_{i=0}^{d-1} \binom{n}{i} \left(q-1\right)^i$.

Comment: Different from the sphere packing bound, which holds for all codes, the sphere covering bound claims the existence of a code. That means, some badly designed codes may not satisfy this bound.

Gilbert-Varshamov Bound: Proof

It's proved by construction.
Let $M_0 = \lceil q^n / \mathrm{Vol}\,(d-1) \rceil > 1$.
It suffices to show that exists a code with M_0 codewords.

Take an arbitrary word $\boldsymbol{c}_1 \in \mathbb{F}_q^n$.
Since $M_0 > 1$, or $q^n > \mathrm{Vol}\,(d-1)$, it holds $\mathbb{F}_q^n \backslash B\,(\boldsymbol{c}_1, d-1) \neq \phi$.
Take an arbitrary word $\boldsymbol{c}_2 \in \mathbb{F}_q^n \backslash B\,(\boldsymbol{c}_1, d-1)$.
It is clear that $d\,(\boldsymbol{c}_1, \boldsymbol{c}_2) \geq d$ $(\boldsymbol{c}_2 \notin B\,(\boldsymbol{c}_1, d-1))$.

Continue this process inductively.
Suppose to obtain codewords $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{M_0-1}$ in this way.
Since Vol $\left(\bigcup_{i=1}^{M_0-1} B\,(\boldsymbol{c}_i, d-1)\right) \leq (M_0 - 1)\,\mathrm{Vol}\,(d-1) < q^n$,
it holds that $\mathbb{F}_q^n \backslash \bigcup_{i=1}^{M_0-1} B\,(\boldsymbol{c}_i, d-1) \neq \phi$.
Take an arbitrary $\boldsymbol{c}_{M_0} \in \mathbb{F}_q^n \backslash \bigcup_{i=1}^{M_0-1} B\,(\boldsymbol{c}_i, d-1) \neq \phi$.

Let $\mathcal{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{M_0}\}$.
By construction, $d\,(\boldsymbol{c}, \boldsymbol{c}') > d - 1$ for all $\boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}$. Hence $d\,(\mathcal{C}) \geq d$. $\Diamond$

Illustration for Sphere Packing and Covering

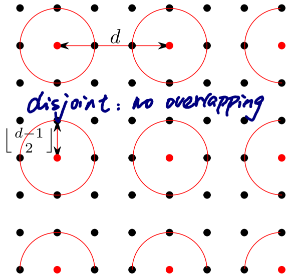
$$\frac{q^n}{\sum_{i=0}^{t}\binom{n}{i}(q-1)^i} \geq M \geq \frac{q^n}{\sum_{i=0}^{d-1}\binom{n}{i}(q-1)^i}$$

$t = \lfloor \frac{d-1}{2} \rfloor$

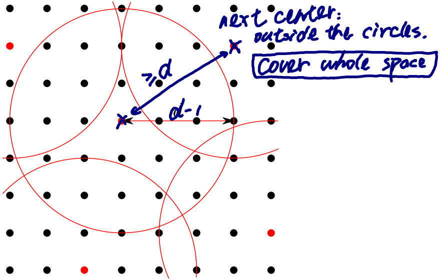
$\text{Volume} = \sum_{i=0}^{d-1}\binom{n}{i}(q-1)^i$

$M \geq \frac{\text{available space}}{\text{ball volume}}$.

Sphere Packing



Sphere Covering



Singleton Bound and MDS

$d-1$ independent

$q^k \le q^{n-d+1}$ $\quad$ $n-k=r$

Theorem 5.4 (Singleton Bound)

$\therefore d \le n-k+1$ $\quad$ $H = \begin{bmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \end{bmatrix}$ $\quad$ $n=7$, $r=3$, $k=4$

The distance of any code $\mathcal{C} \subset \mathbb{F}_q^n$ with M codewords satisfies

$$M \le q^{n-d+1}.$$

In particular, if the code is linear and $M = q^k$, then

$$d \le n-k+1.$$

Definition 5.5 (MDS)

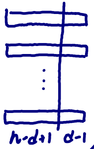
MDS: $d = n-k+1$

Codes that attain the singleton bound are maximum distance separable (MDS).

Binary Hamming codes $\mathcal{H}_r\,[2^r-1, 2^r-1-r, 3]$ are not MDS in general.

- $d = 3 < n-k+1 = r+1$ for all $r \ge 3$.

Singleton Bound: Proof



$(n-d+1, M, \geq 1)$

no repeatition.

$M \leq q^{n-d+1}$

codewords possible combinations

Proof of the general case:
Let $\mathcal{C}$ be of length n and distance d.
$\forall \boldsymbol{c} \in \mathcal{C}$, let $\boldsymbol{c}_{1:n-d+1} \in \mathbb{F}^{n-d+1}$ be the vector containing the first $n-d+1$ entries of $\boldsymbol{c}$, and $\boldsymbol{c}_{n-d+2:n} \in \mathbb{F}^{d-1}$ be the vector composed of the last $d-1$ elements of $\boldsymbol{c}$.

$\forall \boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}$,
$d \leq d_H(\boldsymbol{c}, \boldsymbol{c}') = d_H\left(\boldsymbol{c}_{1:n-d+1}, \boldsymbol{c}'_{1:n-d+1}\right) + d_H\left(\boldsymbol{c}_{n-d+2:n}, \boldsymbol{c}'_{n-d+2:n}\right).$
But $d_H\left(\boldsymbol{c}_{n-d+2:n}, \boldsymbol{c}'_{n-d+2:n}\right) \leq d-1$.
Hence, $d_H\left(\boldsymbol{c}_{1:n-d+1}, \boldsymbol{c}'_{1:n-d+1}\right) \geq d-(d-1) = 1$.
The truncated codewords are all distinct. Hence, $M \leq q^{n-d+1}$. $\Diamond$

Proof for linear codes:
Note that the parity-check matrix $\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$ contains $n-k$ rows.
Every $n-k+1$ columns must be linearly dependent.
By Theorem 4.11, $d \leq n-k+1$. $\Diamond$

Dual of MDS Codes

MDS: $d = n-k+1$

$\mathcal{C}: (n, k, n-k+1)$

$G_c \in \mathbb{F}^{k\times n}$

$H_c \in \mathbb{F}^{(n-k)\times n}$

$\mathcal{C}^\perp: (n, n-k, ?)$ $[k+1]$

$G_c^\perp \in \mathbb{F}^{(n-k)\times n}$

$G_c^\perp = H_c \in \mathbb{F}^{(n-k, n)}$

$d = n-k+1$

$\Rightarrow$ every $d-1 = n-k$ columns of H are linearly independent.

$n-k$ Full rank ($n-k$) | k (H)

Theorem 5.6

If a linear code $\mathcal{C}$ is MDS, then its dual code $\mathcal{C}^\perp$ is also MDS.

$H_c^\perp \in \mathbb{F}^{k\times n}$

each codeword of $\mathcal{C}^\perp$ can be written as $s \in \mathbb{F}^{(1, n-k)}$

$\underbrace{s}_{n-k}$

Let the linear code $\mathcal{C}[n,k]$ be MDS.
According to Theorem 5.6, one has

	Parity-check matrix	Generator Matrix	Parameters
$\mathcal{C}$	$\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$	$\boldsymbol{G} \in \mathbb{F}^{k\times n}$	$(n, k, n-k+1)$
$\mathcal{C}^\perp$	$\boldsymbol{G} \in \mathbb{F}^{k\times n}$	$\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$	$(n, n-k, k+1)$

suppose $d < k+1 \Rightarrow d = k$

$\therefore \exists\, c \in \mathcal{C}^\perp$ s.t. $c \neq 0$ at most k non-zeros, at least $n-k$ zeros (so that $d = k$ possible).

Key for the proof: Theorem 4.11.
If $\mathcal{C}[n,k]$ is MDS, then every set of $n-k$ columns of $\boldsymbol{H}$ is linear independent.

$\because$ first part of H is full rank

$c = sH$

$\therefore$ s is all zero vector.

$\Rightarrow$ c is all zero.

Dual of MDS Codes (Theorem 5.6): Proof

Suppose $d\left(\mathcal{C}^{\perp}\right) < k+1$. Then there exists a nonzero codeword $\boldsymbol{c} \in \mathcal{C}^{\perp}$ with at most k nonzero entries and at least $n-k$ zeros. Since permuting the coordinates reserves the codeword weights (i.e., the distance), w.l.o.g., assume that the last $n-k$ coordinates of $\boldsymbol{c}$ are zeros.

Write the generator matrix of $\mathcal{C}^{\perp}$ (the parity-check matrix of $\mathcal{C}$) as $\boldsymbol{H} = [\boldsymbol{A},\ \boldsymbol{H}']$, where $\boldsymbol{A} \in \mathbb{F}^{(n-k)\times k}$ and $\boldsymbol{H}' \in \mathbb{F}^{(n-k)\times(n-k)}$. By definition of the generator matrix, there exists $\boldsymbol{s} \in \mathbb{F}^{n-k}$ such that $\boldsymbol{c} = \boldsymbol{s}\boldsymbol{H}$.

As $\mathcal{C}$ is MDS, by Theorem 4.11 the columns of $\boldsymbol{H}'$ are linearly independent. That is, $\boldsymbol{H}'$ is invertible. That the last $n-k$ coordinates of $\boldsymbol{c}$ are zeros implies that $\boldsymbol{s} = \boldsymbol{c}_{k+1:n}\left(\boldsymbol{H}'\right)^{-1} = \boldsymbol{0}$. But $\boldsymbol{s} = \boldsymbol{0}$ implies $\boldsymbol{c} = \boldsymbol{s}\boldsymbol{H} = \boldsymbol{0}$ which contradicts the assumption that $\boldsymbol{c} \neq \boldsymbol{0}$. Hence, $d\left(\mathcal{C}^{\perp}\right) \geq k+1$. By the Singleton bound, $d\left(\mathcal{C}^{\perp}\right) = k+1$. $\Diamond$

Section 6
RS & BCH Codes

- Reed-Solomon Codes
  - Definition and properties.
  - Decoding
- Cyclic and BCH codes

The contents in this section are significant re-organization and condensation of the materials of many sources, including Lin & Xing's book, Chapters 7 and 8, and Roth's book, Chapters 5, 6 and 8.

Reed-Solomon Codes

Our Heroes: Irving S. Reed and Gustave Solomon

Used in

- Magnetic recording (all computer hard disks use RS codes)
- Digital versatile disks (CDs, DVDs, etc.)
- Optical fiber networks (ITU-TG.795)
- ADSL transceivers (ITU-TG.992.1)
- Wireless telephony (3G systems, 4G systems)
- Digital satellite broadcast (ETS 300-421S, ETS 300-429)
- Deep space exploration (all NASA probes)

RS Codes: Evaluation Mapping

Definition 6.1 (Evaluation Mapping)

Let $\mathbb{F}_q$ be a finite field. Let $n \leq q-1$ (typically $n = q-1$).
Let $\mathcal{A} = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_q$.
For any polynomial $f(x) \in \mathbb{F}_q[x]$, define the evaluation mapping eval $(f(x))$ that maps f to a vector $\boldsymbol{c} \in (\mathbb{F}_q)^n$

$\mathbb{F}_7 = \{0, 1 \cdots 6\}$. $A = \{1, \alpha \cdots \alpha^5\} = \{1.3.2.6.4.5\}$ $\alpha = 3$

$$f \mapsto \boldsymbol{c} = [c_1, \cdots, c_n] \text{ where } c_i = f(\alpha_i).$$

① $f_1(x) = 2x+1$

$C = \text{eval}(f_1) = \{\underbrace{2\times1+1}_{=3}, \underbrace{2\times3+1}_{=0}, \underbrace{2\times2+1}_{=5}, 6.2.4\}$

$c_i = f(\alpha_i)$

Example 6.2

$\mathbb{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$. Choose the primitive element $\alpha = 3$.
Let $\mathcal{A} = \{1, \alpha, \cdots, \alpha^5\} = \{1, 3, 2, 6, 4, 5\}$.
$f(x) = 2x + 1$, $\boldsymbol{c} = \text{eval}(f) = [3, 0, 5, 6, 2, 4]$.
$f(x) = 3x^2 + x + 2$, $\boldsymbol{c} = \text{eval}(f) = [6, 4, 2, 4, 5, 5]$.

② $f_2(x) = 3x^2 + x + 2$

$c = \text{eval}(f_2) = \{\underbrace{3\times1^2+1+2}_{=6}, \underbrace{3\times3^2+3+2}_{=4}, 2.4.5.5\}$

RS Codes: Definition

$\mathbb{F}_q^{[x]}$ degree $k-1$

$f(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$, $a_i \in \mathbb{F}_q$.

$|f(x)| = q^k$

$\mathbb{F}_q = \{0, 1 \ldots p-1 \ldots\} = \{0, 1, \alpha \cdots \alpha^{q-2}\}$

$\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\} = \{1, \alpha, \ldots \alpha^{q-2}\}$

$\Downarrow c_i = f(\alpha_i)$

$C = \{[f(1), f(\alpha) \cdots f(\alpha^{q-2})], \deg(f) = k-1\}$

Definition 6.3 (Reed-Solomon Codes)

Given $\mathcal{A} = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_q$, an $[n, k]$ q-ary RS code $\mathcal{C} = \{\text{eval}(f),\ 0 \le \deg f \le k-1\}$.

The set $\mathcal{A}$ is called a defining set of points of $\mathcal{C}$.

$C\ (q-1,\ k,\ ? \to n-k+1)$

$= q-k$ ~ distance (MDS)

length dimension

A common choice of defining set of points of $\mathcal{C}$ is $\mathcal{A} = \{1, \alpha, \cdots, \alpha^{q-2}\}$ where α is a primitive element in $\mathbb{F}_q$.

In this case, $n = q - 1$.

RS Codes: Properties

Theorem 6.4

1. *RS codes are linear codes.*
2. *RS codes are MDS, i.e., The distance of the RS code is* $d = n - k + 1$.

Proof:

1. Let $\boldsymbol{c}_1 = \text{eval}\,(f_1)$ and $\boldsymbol{c}_2 = \text{eval}\,(f_2)$ where deg $f_1 \leq k-1$ and deg $f_2 \leq k-1$. Then $\alpha \boldsymbol{c}_1 + \beta \boldsymbol{c}_2 = \text{eval}\,(g)$ with $g = \alpha f_1 + \beta f_2$. Since deg $g \leq k-1$, $\text{eval}\,(g) \in \mathcal{C}$.
2. A polynomial of degree $\leq k-1$ can have at most $k-1$ zeros. Hence, $\forall \boldsymbol{c} \in \mathcal{C}$ s.t. $\boldsymbol{c} \neq 0$, $\boldsymbol{c} = \text{eval}\,(f)$ has weight at least $n-k+1$. $\Diamond$

polynomial of degree k-1 ⇒ at most k-1 zeros for each codeword

⇒ non-zero element at least n-k+1

⇒ distance = n-k+1

RS Codes: Conventional Definition

$f = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$

$f(1) = a_0 + a_1 + \cdots + a_{k-1}$

$f(\alpha) = a_0 + a_1\alpha + \cdots + a_{k-1}\alpha^{k-1}$

$\vdots$

$f(\alpha^{n-1}) = a_0 + a_1\alpha^{n-1} + \cdots + a_{k-1}\alpha^{(k-1)(n-1)}$

$C = S \; G$

$[f(1).f(\alpha)\cdots f(\alpha^{n-1})]$ $\quad$ $[a_0.a_1\cdots a_{k-1}]$

Theorem 6.5

Let the defining set of points is $\{1, \alpha, \cdots, \alpha^{n-1}\}$ *with* $\mathrm{order}(\alpha) = n$ *(typically* $n = q - 1$*). The generated RS code has generator matrix and parity-check matrix given by*

$$\underset{k\times n}{\boldsymbol{G}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \cdots & \alpha^{(k-1)(n-1)} \end{bmatrix}$$

and

$$\underset{(n-k)\times n}{\boldsymbol{H}} = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \cdots & \alpha^{(n-k)(n-1)} \end{bmatrix}$$

Generator Matrix: Justification

For any $\boldsymbol{c} \in \mathcal{C}$, there exists a polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$ s.t. $\boldsymbol{c} = \text{eval}(f) = \left[f(1), f(\alpha), \cdots, f\left(\alpha^{n-1}\right) \right] \in \mathcal{C}$.

Note that $\forall i \in [n]$, $c_i = f\left(\alpha^i\right) = \sum_{\ell=0}^{k-1} a_\ell \left(\alpha^{i-1}\right)^\ell = [a_0, \cdots, a_{k-1}] \boldsymbol{G}_i$ where $\boldsymbol{G}_i$ is the i-th column of the $\boldsymbol{G}$ matrix.

One has $\mathcal{C} = \left\{ \boldsymbol{s}\boldsymbol{G} : \ \boldsymbol{s} \in \mathbb{F}_q^k \right\}$ and $\boldsymbol{G}$ is a generator matrix of $\mathcal{C}$.

Remark: In the definition of the generator matrix (Def. 4.5), the rows of $\boldsymbol{G}$ are required to be linearly independent. We shall prove it later.

Parity-Check Matrix: Justification

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \cdots & \alpha^{(k-1)(n-1)} \end{bmatrix} \; i$$

$A_{ij} = \sum (i^{th} \text{ row of } G)\cdot(j^{th} \text{ row of } H)$

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

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

Lemma 6.6

The $\boldsymbol{G}$ and $\boldsymbol{H}$ matrices defined in Theorem 6.5 satisfy $\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$.

L

$= \sum_{l=1}^{n} \alpha^{(i+j-1)(l-1)} = \dfrac{\alpha^{(i+j-1)n} - 1}{\alpha^{(i+j-1)} - 1} = 0$

Proof: Let $\boldsymbol{A} := \boldsymbol{G}\boldsymbol{H}^T \in \mathbb{F}_q^{k\times(n-k)}$.
$\forall i \in [k]$ and $\forall j \in [n-k]$, it holds that

$$\boldsymbol{A}_{i,j} = \sum_{\ell=1}^{n} \alpha^{(\ell-1)(i-1)}\alpha^{(\ell-1)j} = \sum_{\ell=1}^{n} \alpha^{(i+j-1)(\ell-1)}$$

$$\overset{(a)}{=} \frac{\alpha^{(i+j-1)n} - 1}{\alpha^{i+j-1} - 1} \overset{(b)}{=} 0,$$

where (a) comes from that $i+j-1 < n$ and $\alpha^{i+j-1} \neq 1$, and (b) holds because $\alpha^n = 1$.

Row Rank of the $\boldsymbol{G}/\boldsymbol{H}$ Matrix

Theorem 6.7

The rows of the $\boldsymbol{G}/\boldsymbol{H}$ matrix in Theorem 6.5 are linearly independent.

Proof: It is sufficient to show that any k-column sub-matrix of $\boldsymbol{G}$ ($(n-k)$-column sub-matrix of $\boldsymbol{H}$) has full rank.

Note that a k-column sub-matrix of $\boldsymbol{G}$ is of the form

$$\boldsymbol{G}' = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{i_k} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(k-1)i_1} & \alpha^{(k-1)i_2} & \cdots & \alpha^{(k-1)i_k} \end{bmatrix},$$

which is a Vandermonde matrix (defined and analysed later). A Vandermonde matrix has full rank. Hence the rows of $\boldsymbol{G}$ are linearly independent. $\Diamond$

Vandermonde Matrix

Definition 6.8 (Vandermonde Matrix)

A Vandermonde matrix $\boldsymbol{V} \in \mathbb{F}^{n \times n}$ is of the form

$$\boldsymbol{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-1} & \alpha_2^{n-1} & \cdots & \alpha_n^{n-1} \end{bmatrix}.$$

Theorem 6.9

The determinant of a Vandermonde matrix $\boldsymbol{V} \in \mathbb{F}^{n \times n}$ *is*

$$|\boldsymbol{V}| = \prod_{i<j} (\alpha_j - \alpha_i).$$

As a result, if $\alpha_i \neq \alpha_j$, $1 \leq i \neq j \leq n$, *then* $|\boldsymbol{V}| \neq 0$ *and* $\boldsymbol{V}$ *is of full rank.*

Determinant: A Recap

Definition 6.10 (Determinant)

$\forall \boldsymbol{A} \in \mathbb{F}^{n\times n}$, its determinant $|\boldsymbol{A}|$ is computed via

$$|\boldsymbol{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{i,j} |\boldsymbol{M}_{i,j}|,$$

where $\boldsymbol{M}_{i,j}$ is the minor matrix obtained by deleting row i and column j from $\boldsymbol{A}$.

Lemma 6.11

1. $|\boldsymbol{AB}| = |\boldsymbol{A}|\,|\boldsymbol{B}|$.
2. *If $\boldsymbol{B}$ results from $\boldsymbol{A}$ by adding a multiple of one row/column to another row/column, then $|\boldsymbol{B}| = |\boldsymbol{A}|$.*
3. $|\boldsymbol{A}| \neq 0 \Leftrightarrow \boldsymbol{A}$ *is of full rank.*

Theorem 6.9: Proof (1)

We prove Theorem 6.9 by using induction.

Recall that

$$\boldsymbol{V}_n = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & \alpha_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{n-2} & \alpha_2^{n-2} & \cdots & \alpha_{n-1}^{n-2} & \alpha_n^{n-2} \\ \alpha_1^{n-1} & \alpha_2^{n-1} & \cdots & \alpha_{n-1}^{n-1} & \alpha_n^{n-1} \end{bmatrix}$$

Let $(\boldsymbol{V}_n')_{:,2} = (\boldsymbol{V}_n)_{:,2} - (\boldsymbol{V}_n)_{:,1}, \cdots, (\boldsymbol{V}_n')_{:,n} = (\boldsymbol{V}_n)_{:,n} - (\boldsymbol{V}_n)_{:,1}$. We obtain

minus first column

$$\boldsymbol{V}_n' = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ \alpha_1 & \alpha_2 - \alpha_1 & \cdots & \alpha_{n-1} - \alpha_1 & \alpha_n - \alpha_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{n-2} & \alpha_2^{n-2} - \alpha_1^{n-2} & \cdots & \alpha_{n-1}^{n-2} - \alpha_1^{n-2} & \alpha_n^{n-2} - \alpha_1^{n-2} \\ \alpha_1^{n-1} & \alpha_2^{n-1} - \alpha_1^{n-1} & \cdots & \alpha_{n-1}^{n-1} - \alpha_1^{n-1} & \alpha_n^{n-1} - \alpha_1^{n-1} \end{bmatrix}$$

$$\alpha_2^{n-1} - \alpha_1^{n-1} - \alpha_1(\alpha_2^{n-2} - \alpha_1^{n-2})$$

$$= \alpha_2^{n-1} - \alpha_1 \alpha_2^{n-2} = \alpha_2^{n-2}(\alpha_2 - \alpha_1)$$

Theorem 6.9: Proof (2)

Let $(V_n'')_{n,:} = (V_n')_{n,:} - \alpha_1 (V_n')_{n-1,:}, \cdots, (V_n'')_{2,:} = (V_n')_{2,:} - \alpha_1 (V_n')_{1,:}$. We obtain

minus (last row $\cdot \alpha_1$)

$$V_n'' = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \alpha_2 - \alpha_1 & \cdots & \alpha_{n-1} - \alpha_1 & \alpha_n - \alpha_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \alpha_2^{n-2} - \alpha_1 \alpha_2^{n-3} & \cdots & \alpha_{n-1}^{n-2} - \alpha_1 \alpha_{n-1}^{n-3} & \alpha_n^{n-2} - \alpha_1 \alpha_n^{n-3} \\ 0 & \alpha_2^{n-1} - \alpha_1 \alpha_2^{n-2} & \cdots & \alpha_{n-1}^{n-1} - \alpha_1 \alpha_{n-1}^{n-2} & \alpha_n^{n-1} - \alpha_1 \alpha_n^{n-2} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \alpha_2 - \alpha_1 & \cdots & \alpha_{n-1} - \alpha_1 & \alpha_n - \alpha_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & (\alpha_2 - \alpha_1)\, \alpha_2^{n-3} & \cdots & (\alpha_{n-1} - \alpha_1)\, \alpha_{n-1}^{n-3} & (\alpha_n - \alpha_1)\, \alpha_n^{n-3} \\ 0 & (\alpha_2 - \alpha_1)\, \alpha_2^{n-2} & \cdots & (\alpha_{n-1} - \alpha_1)\, \alpha_{n-1}^{n-2} & (\alpha_n - \alpha_1)\, \alpha_n^{n-2} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \\ & V_{n-1}(\alpha_2, \cdots, \alpha_n) \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & \alpha_2 - \alpha_1 & & & \\ & & \ddots & & \\ & & & \alpha_{n-1} - \alpha_1 & \\ & & & & \alpha_n - \alpha_1 \end{bmatrix}$$

Hence $|V_n| = |V_n'| = |V_n''| = |V_{n-1}| \prod_{j>1} (\alpha_j - \alpha_1)$. ◊

Decoding with Known Error Locations

Let $\boldsymbol{e}$ be the error vector.
Let $\mathcal{I} = \{i : \ e_i \neq 0\}$ be the set of error locations.
$\boldsymbol{e}_{\mathcal{I}}, \ \boldsymbol{H}_{\mathcal{I}}$: sub-vector and sub-matrix of $\boldsymbol{e}$ and $\boldsymbol{H}$ respectively.

$H_I y_I^T = H_I e_I^T = s^T$

If we knew error locations $\mathcal{I}$:

$$\text{Solve } \underbrace{\boldsymbol{H}_{\mathcal{I}}}_{\text{full rank}} \boldsymbol{e}_{\mathcal{I}}^T = \boldsymbol{s}^T. \ (\boldsymbol{e}_{\mathcal{I}}^T = \boldsymbol{H}_{\mathcal{I}}^{\dagger} \boldsymbol{s}^T)$$

Complexity of pseudo-inverse $\boldsymbol{H}_{\mathcal{I}}^{\dagger}$: $O\left(d^3\right)$.

Erasure Correction

Recall the erasure channel model.
Suppose that $\boldsymbol{c} \in \mathcal{C}$ was transmitted.
Receive $\boldsymbol{r} = [c_1 \cdots c_{i-1} \ ? \ c_{i+1} \cdots c_n]$ (at most $d-1$ symbols erased).

Decoding: Set the missing symbols to zero, i.e., $\boldsymbol{r}_{\mathcal{I}} = \boldsymbol{0}$.
Then $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e}_{\mathcal{I}^c} = \boldsymbol{0}$.
$\boldsymbol{s}^T = \boldsymbol{H}\boldsymbol{r}^T = \boldsymbol{H}\boldsymbol{e}^T = \boldsymbol{H}_{\mathcal{I}}\boldsymbol{e}_{\mathcal{I}}^T$.

$$\begin{bmatrix} \alpha^{i_1-1} & \alpha^{i_2-1} & \cdots & \alpha^{i_s-1} \\ \alpha^{2(i_1-1)} & \alpha^{2(i_2-1)} & \cdots & \alpha^{2(i_s-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{s(i_1-1)} & \alpha^{s(i_2-1)} & \cdots & \alpha^{s(i_s-1)} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_s \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_s \end{bmatrix}.$$

A Specific Example

Consider a $[7,4,4]$ RS code over $\mathbb{F}_8$ ($\mathbb{F}_2[x]/x^3+x+1$).
Let α be a primitive element (a root of $f(x) = x^3+x+1$).

$\alpha \cdot 1^3 + \alpha \cdot 1^2 + 1$

$\mathbb{F}_8$: $\alpha^3 = \alpha + 1$

000	001	010	100	011	110	111	101
0	1	α	α^2	α^3	α^4	α^5	α^6

Encoded message $m(x) = \alpha x^3 + \alpha x^2 + x$.
$\boldsymbol{c} = \mathrm{eval}(m) = \begin{bmatrix} 1 & \alpha^5 & \alpha & 1 & \alpha^5 & \alpha^6 & \alpha^5 \end{bmatrix}$.
$\boldsymbol{r} = \begin{bmatrix} 1 & \alpha^5 & \alpha & 1 & ? & ? & \alpha^5 \end{bmatrix}$.

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

$$\boldsymbol{c}\boldsymbol{H}^T = \boldsymbol{0} \Rightarrow \begin{bmatrix} \alpha^4 & \alpha^5 \\ \alpha & \alpha^3 \\ \alpha^5 & \alpha \end{bmatrix} \begin{bmatrix} c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} \alpha \\ 1 \\ \alpha \end{bmatrix} \overset{\text{matrix inverse}}{\Longrightarrow} \begin{bmatrix} c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} \alpha^5 \\ \alpha^6 \end{bmatrix}.$$

Error Correction

In previous example:

- "Error" (erasure) locations are known.
- The error values are found via matrix inverse.

For error correction:

- Find error locations
  - Exhaustive search: complexity $\binom{n}{t} = O\left(n^t\right)$.
  - In practice, $\exists$ methods to find error locations efficiently.
- Correct errors with given error locations
  - Methods to avoid matrix inverse.

Efficient Error Correction: Definitions

Definitions 6.12

Syndrome polynomial:

$$S(z) = \sum_{j=0}^{n-k-1} s_j z^j, \text{ where } \boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T.$$

Error locator polynomial: (information about error locations)

$$L(z) = \prod_{i \in \mathcal{I}} \left(1 - \alpha^i z\right).$$

Error evaluator polynomial: (information about errors)

$$E(z) = L(z) \sum_{i \in \mathcal{I}} \frac{e_i \alpha^i}{(1-\alpha^i z)} = \sum_{i \in \mathcal{I}} e_i \alpha^i \prod_{j \in \mathcal{I} \setminus \{i\}} \left(1 - \alpha^j z\right).$$

Remark: The receiver can compute the syndrome vector and the syndrome polynomial easily.

Information Encoded in $L(z)$ and $E(z)$

- If we know $L(z)$, we can find error locations.
  - $\begin{cases} L\left(\alpha^{-k}\right) = 0 & \text{if } k \in \mathcal{I} \\ L\left(\alpha^{-k}\right) \neq 0 & \text{if } k \notin \mathcal{I} \end{cases}.$
  - Error locations can be found by exhaustively computing $L\left(\alpha^{-k}\right)$, $0 \le k \le n-2$.

- $E(z)$ helps find errors e_i, $i \in \mathcal{I}$, without matrix inverse.
  - $\forall k \in \mathcal{I}$, $E\left(\alpha^{-k}\right) = e_k \alpha^k \prod_{j \neq k} \left(1 - \alpha^j \alpha^{-k}\right) \neq 0$.
  - $e_k = E\left(\alpha^{-k}\right) / \left(\alpha^k \prod_{j \neq k} \left(1 - \alpha^j \alpha^{-k}\right)\right)$.
  - or $e_k = -E\left(\alpha^{-k}\right) / \frac{d}{dz} L\left(\alpha^{-k}\right)$, where $\frac{d}{dz} L(z)$ is the derivative of $L(z)$.
  - Complexity is reduced from $O\left(n^3\right)$ to $O\left(n^2\right)$.

Decoding strategy: from $S(z)$ to find $L(z)$ and $E(z)$.

An Example of $L(z)$ and $E(z)$

- $\mathcal{I} = \{1, 2, 5\}$.
- $L(z) = (1 - \alpha z)\left(1 - \alpha^2 z\right)\left(1 - \alpha^5 z\right)$
  - $L(z) = 0$ if $z = \alpha^{-1}, \alpha^{-2},$ or α^{-5}.
  - $L(z) \neq 0$ otherwise.
- $$\begin{aligned} E(z) = \quad & e_1 \alpha^1 \left(1 - \alpha^2 z\right)\left(1 - \alpha^5 z\right) && \text{T1} \\ & + e_2 \alpha^2 (1 - \alpha z)\left(1 - \alpha^5 z\right) && \text{T2} \\ & + e_5 \alpha^5 (1 - \alpha z)\left(1 - \alpha^2 z\right) && \text{T3} \end{aligned}$$

	T1	T2	T3	$E(z)$
$z = \alpha^{-1}$	$\neq 0$	$= 0$	$= 0$	$\neq 0$
$z = \alpha^{-2}$	$= 0$	$\neq 0$	$= 0$	$\neq 0$
$z = \alpha^{-5}$	$= 0$	$= 0$	$\neq 0$	$\neq 0$

Properties of $L(z)$ and $E(z)$

Let $t = \left\lfloor \frac{d-1}{2} \right\rfloor$.

Theorem 6.13

1. $\gcd\left(L(z), E(z)\right) = 1$.
2. *The key equation:*

$$E(z) = L(z)\, S(Z) \mod z^{d-1}.$$

3. *(Uniqueness) Let $a(z), b(z) \in \mathbb{F}_q[z]$ be such that deg$\left(a(z)\right) \le t-1$, deg$\left(b(z)\right) \le t$, gcd$\left(a(z), b(z)\right) = 1$ and*

$$a(z) \equiv S(z)\, b(z) \ \left(\mathit{mod}\ z^{d-1}\right).$$

Then $a(z)$ and $b(z)$ are unique up to a constant.
That is, we can treat $a(z) = cE(z)$, $b(z) = cL(z)$, and $E(z)$ and $L(z)$ are generated from an error vector $\boldsymbol{e}$ s.t. $\mathrm{wt}(\boldsymbol{e}) \le t$.

Decoding Process

1. Compute the syndrome vector and polynomial $\boldsymbol{s}$ and $S(z)$ respectively.
2. Apply Euclidean algorithm to z^{d-1} and $S(z)$, i.e.,
$$\begin{aligned} z^{d-1} &= q_1(z) S(z) + r_1(z) \\ S(z) &= q_2(z) r_1(z) + r_2(z) \\ &\vdots \\ r_{\ell-2}(z) &= q_\ell(z) r_{\ell-1}(z) + r_\ell(z), \end{aligned}$$
where $\deg(r_\ell(z)) \leq t-1$.
3. By Bézout's Identity (Lem. 1.5), one has
$$r_\ell(z) = a(z) S(z) + b(z) z^{d-1} \equiv a(z) S(z) \mod z^{d-1}.$$
4. Let c be the leading coefficient of the polynomial $a(z)$, i.e., $c^{-1}a(z)$ is a monic polynomial. By Theorem 6.13, set
$$L(z) = c^{-1}a(z), \text{ and } E(z) = c^{-1}r_\ell(z).$$
5. Find the error locations $i \in \mathcal{I}$ from $L(z)$ and the errors e_i from $E(z)$.
$$\hat{\boldsymbol{c}} = \boldsymbol{y} - \boldsymbol{e}.$$

The complexity is highly reduced!

Theorem 6.13, Part 1: Proof

Proof: $L(z)$ has roots α^{-i}, $i \in \mathcal{I}$. None of them is a root of $E(z)$.
$L(z)$ and $E(z)$ does not share any roots.
$\gcd(L(z), E(z)) = 1$.

Theorem 6.13, Part 2: Proof

Theorem 6.13 part 2 is a direct consequence of the lemma below.

Lemma 6.14

$S(z) \equiv \sum_{i \in \mathcal{I}} \frac{e_i \alpha^i}{1-\alpha^i z} \mod z^{d-1}$

Proof: As $\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$, it follows that
Hence, $s_j = \sum_{i=0}^{n-1} e_i \alpha^{i(j+1)} = \sum_{i \in \mathcal{I}} e_i \alpha^{i(j+1)}$, $\forall 0 \le j \le d-2$.

By the definition of $S(z)$, it holds that

$$\begin{aligned} S(z) &= \sum_{j=0}^{d-2} s_j z^j = \sum_{j=0}^{d-2} \sum_{i \in \mathcal{I}} e_i \alpha^{i(j+1)} z^j \\ &= \sum_{i \in \mathcal{I}} e_i \alpha^i \left(\sum_{j=0}^{d-2} \left(\alpha^i z\right)^j \right) \\ &= \sum_{i \in \mathcal{I}} e_i \alpha^i \left(\sum_{j=0}^{\infty} \left(\alpha^i z\right)^j \right) \mod z^{d-1} \\ &= \sum_{i \in \mathcal{I}} e_i \alpha^i \frac{1}{1-\alpha^i z}. \end{aligned}$$

$\Diamond$

Theorem 6.13, Part 3: Proof

Proof: To prove the uniqueness, we assume that there exist $\left(E\left(z\right),L\left(z\right)\right)\neq\left(E'\left(z\right),L'\left(z\right)\right)$ s.t.

$E\left(z\right)=S\left(z\right)L\left(z\right) \mod z^{d-1}$ and $E'\left(z\right)=S\left(z\right)L'\left(z\right) \mod z^{d-1}$.

It follows that

$$\begin{aligned} E\left(z\right)L'\left(z\right) &= S\left(z\right)L\left(z\right)L'\left(z\right) \mod z^{d-1} \\ &= E'\left(z\right)L\left(z\right) \mod z^{d-1}. \end{aligned} \tag{2}$$

By assumption, $\deg\left(E\left(z\right)\right)\leq t-1$ and $\deg\left(L'\left(z\right)\right)\leq t$.
It is clear that $\deg\left(E\left(z\right)L'\left(z\right)\right)\leq 2t-1\leq d-2$.
The same is true for $E'\left(z\right)L\left(z\right)$.
As a result, (2) becomes

$$E\left(z\right)L'\left(z\right)=E'\left(z\right)L\left(z\right)$$

Note gcd $\left(E\left(z\right),L\left(z\right)\right)=1$. By Lemma 1.12, $E\left(z\right)|E'\left(z\right)$ and $L\left(z\right)|L'\left(z\right)$.
Similarly from gcd $\left(E'\left(z\right),L'\left(z\right)\right)=1$, $E'\left(z\right)|E\left(z\right)$ and $L'\left(z\right)|L\left(z\right)$.
Hence, $E\left(z\right)=cE'\left(z\right)$ and $L\left(z\right)=cL'\left(z\right)$ for some nonzero $c\in\mathbb{F}_q$. $\Diamond$

An Example

Example: Consider the $[7,3]$ RS code over $\mathbb{F}_8$ ($\mathbb{F}_8$ is given as follows).

0	1	α	α^2	α^3	α^4	α^5	α^6
000	001	010	100	011	110	111	101

Let the received signal be $\boldsymbol{y} = \left[\alpha^3, \alpha, 1, \alpha^2, 0, \alpha^3, 1\right]$. Find $\hat{\boldsymbol{c}}$.

Solutions to the Example

1. Parameters: $n-k=4$, $d=5$ $(t=2)$, and $\boldsymbol{H} \in \mathbb{F}_8^{4\times 7}$.
$$\boldsymbol{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \\ 1 & \alpha^4 & \alpha & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \end{bmatrix}.$$
2. Syndromes: $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T = \left[\alpha^3, \alpha^4, \alpha^4, 0\right]$.
$$S(z) = \alpha^4 z^2 + \alpha^4 z + \alpha^3.$$
3. Key polynomials: apply Euclidean algorithm to z^4 and $S(z)$.
   - 3.1 $z^4 = \left(\alpha^3 z^2 + \alpha^3 z + \alpha^5\right) S(z) + (z+\alpha)$.
   - 3.2 $L'(z) = \alpha^3 z^2 + \alpha^3 z + \alpha^5$. $E'(z) = z + \alpha$.
   - 3.3 $L(z) = \alpha^5 z^2 + \alpha^5 z + 1$. $E(z) = \alpha^2 z + \alpha^3$.
4. Find $\hat{\boldsymbol{c}}$:
   - 4.1 Plug $1, \alpha^{-1}, \cdots$ into $L(z)$. $L\left(\alpha^{-2}\right) = L\left(\alpha^{-3}\right) = 0$.
   - 4.2 According $E(z)$, we have $e_2 = \alpha^3$ and $e_3 = \alpha^6$.
   - 4.3 $\hat{\boldsymbol{c}} = \boldsymbol{y} - \boldsymbol{e} = \boldsymbol{y} + \boldsymbol{e} = \left[\alpha^3, \alpha, \alpha, 1, 0, \alpha^3, 1\right]$.

Towards Cyclic and BCH Codes

Have seen

- Binary Hamming codes: $d = 3$.
- Reed-Solomon codes: MDS $(d = n - k + 1)$ and requires large fields (typically $q = n + 1$).

Will introduce cyclic codes

- Reed-Solomon codes are a special case of cyclic codes.
- BCH codes as another special case.
  - Systematic way to construct binary codes with large distance.

Cyclic Codes

Definition 6.15

An $[n, k]$ *linear* code is cyclic
if for every codeword $\boldsymbol{c} = c_0 c_1 \cdots c_{n-2} c_{n-1}$,
the right cyclic shift of $\boldsymbol{c}$, $c_{n-1} c_0 c_1 \cdots c_{n-2}$, is also a codeword.

Example: The $\mathcal{H}\,[7, 3]$ has the parity-check matrix

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

It's dual code $\mathcal{H}_3^{\perp}$ (view $\boldsymbol{H}$ as the generator matrix) is cyclic.
(The codewords are 1011100, 0101110, 0010111, 1110010, 1001011, 0111001, 1100101, 0000000.)

Generating Function

Definition 6.16

The generating function of a codeword $\boldsymbol{c} = [c_0 \cdots c_{n-1}]$ is $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$.

It will be convenient to use $c(x)$ to represent a codeword $\boldsymbol{c}$.

The right cyclic shift $\boldsymbol{c} = (c_0, \cdots, c_{n-1}) \mapsto \boldsymbol{c}' = (c_{n-1}, c_0, \cdots, c_{n-2})$ can be obtained by $c'(x) = x \cdot c(x) \mod x^n - 1$ as

$$\begin{aligned} x \cdot c(x) &= c_0 x + c_1 x^2 + \cdots + c_{n-2} x^{n-1} + c_{n-1} x^n \\ &= c_{n-1} + c_0 x + c_1 x^2 + \cdots + c_{n-2} x^{n-1} \bmod x^n - 1. \end{aligned}$$

Lemma 6.17

Let $c(x) \in \mathcal{C}$. For an arbitrary $u(x)$, $u(x)\, c(x) \mod x^n - 1$ is in $\mathcal{C}$.

Generator Polynomial

Theorem 6.18

For a cyclic code $\mathcal{C}$, $\exists$ a unique monic polynomial $g(x)$ s.t. for all $c(x) \in \mathcal{C}$, $c(x) = u(x)\, g(x)$ for some $u(x)$.

Proof:

Let $g(x) \in \mathcal{C}$ be the nonzero polynomial of least degree.
Since $\mathcal{C}$ is linear, w.l.o.g., assume that $g(x)$ is monic.
Then $\forall c(x) \in \mathcal{C}$, write $c(x) = u(x)\, g(x) + r(x)$.
By definition of cyclic codes, $u(x)\, g(x) \in \mathcal{C}$.
Hence, $r(x) \in \mathcal{C}$ by linearity of $\mathcal{C}$.
But $\deg(r(x)) < \deg(g(x))$, which implies $r(x) = 0$.

The uniqueness of $g(x)$ can be proved by contradiction. Suppose that there are two *monic* polynomials $g_1(x) \neq g_2(x)$ of the same degree that both generate $\mathcal{C}$. Then $g_1(x) - g_2(x) \in \mathcal{C}$ and $\deg(g_1 - g_2) < \deg(g_1)$, which forces $g_1(x) - g_2(x) = 0$. $\Diamond$

Properties of the Generator Polynomial

Corollary 6.19

$g(x) \,|\, x^n - 1$.

Proof: Write $x^n - 1 = q(x)\,g(x) + r(x)$.
Take "$\mathrm{mod}\ x^n - 1$" on both sides.
$0 = x^n - 1 \bmod x^n - 1 \in \mathcal{C}$. $q(x)\,g(x) \bmod x^n - 1 \in \mathcal{C}(x)$.
Hence $r(x) \bmod x^n - 1 \in \mathcal{C} \Rightarrow r(x) \in \mathcal{C} \Rightarrow r(x) = 0$. $\Diamond$

Remark: Let $n = q^m - 1$.
We know how to factor $x^n - 1$ in terms of minimal polynomials.
$g(x)$ must be a product of minimal polynomials.

Generator Matrices of Cyclic Codes

Theorem 6.20

The generator matrix of a cyclic code $\mathcal{C}[n,k]$:

$$\boldsymbol{G} = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & & & \\ & g_0 & g_1 & \cdots & g_{n-k} & & \\ & & \ddots & & & \ddots & \\ & & & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}.$$

Observations:

- $\deg(g(x)) = n-k$.
- Easy for implementation: can be implemented by using flip-flops.

Parity-Check Matrices of Cyclic Codes

Recall $g(x) \mid x^n - 1$. Define $h(x)$ such that $g(x)\,h(x) = x^n - 1$.
Then $h(x)$ is a *monic* polynomial with *degree* k.

Write $h(x) = \sum_{i=0}^{k} a_i x^i$.

Definition 6.21

The reciprocal polynomial $h_R(x)$ of $h(x)$ is given by

$$h_R(x) = a_k + a_{k-1}x + \cdots + a_0 x^k = x^k h(1/x).$$

Example: $h(x) = 1 + x^2 + x^3 \Rightarrow h_R(x) = 1 + x + x^3$.

Parity-Check Matrix

Theorem 6.22

The parity-check matrix of the cyclic code $\mathcal{C}[n,k]$ is

$$\boldsymbol{H} = \begin{bmatrix} h_R(x) \\ x h_R(x) \\ \vdots \\ x^{n-k-1} h_R(x) \end{bmatrix} = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & & & & \\ & h_k & h_{k-1} & \cdots & h_0 & & & \\ & & \ddots & & & \ddots & & \\ & & & h_k & h_{k-1} & \cdots & h_0 \end{bmatrix}.$$

Corollary 6.23

The dual of a cyclic code, $\mathcal{C}^{\perp}$, is also cyclic.

$h_0^{-1} h_R(x)$ is the generator polynomial of $\mathcal{C}^{\perp}$.

Theorem 6.22: Proof

By assumption, $x^n - 1 = g(x)\,h(x)$. Note that

$$\begin{aligned} g(x)\,h(x) &= \left(\sum_{i=0}^{n-k} g_i x^i\right)\left(\sum_{i=0}^{k} h_i x^i\right) \\ &= \sum_{i=0}^{n}\left(\sum_{\ell=0}^{i} g_\ell h_{i-\ell}\right) x^i = \sum_{i=0}^{n} a_i x^i, \end{aligned}$$

where $a_0 = g_0 h_0 = -1$, $a_n = g_{n-k} h_k = 1 \cdot 1 = 1$, and

$$a_i = \sum_{\ell=0}^{i} h_\ell g_{i-\ell} = 0, \quad 1 \le i \le n-1.$$

Let $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{H}^T$ with

$$\boldsymbol{A}_{i,j} = [\underbrace{0,\cdots,0}_{i-1}, g_0, \cdots, g_{n-k}, 0, \cdots 0] \cdot [\underbrace{0,\cdots,0}_{j-1}, h_k, \cdots, h_0, 0, \cdots, 0]^T.$$

It can be verified that $A_{1,1} = a_k$, $A_{1,2} = a_{k+1}$, $\cdots$, and

$$\boldsymbol{A} = \boldsymbol{G}\boldsymbol{H}^T = \begin{bmatrix} a_k & a_{k+1} & \cdots & a_{n-1} \\ a_{k-1} & a_k & \cdots & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \cdots & a_{n-k} \end{bmatrix} = \boldsymbol{0} \in \mathbb{F}^{k\times(n-k)} \qquad \Diamond$$

Cyclic Codes: An Example

To construct a cyclic code on $\mathbb{F}_q$, we realize that

- $M^{(i)}(x) \in \mathbb{F}_q[x]$
- $M^{(i)}(x) | x^{q^m-1} - 1$.

Definition 6.24

A BCH code over $\mathbb{F}_q$ of length $n = q^m - 1$ is the cyclic code generated by

$$g(x) = \text{lcm}\left(M^{(a)}(x), \cdots, M^{(a+\delta-2)}(x)\right)$$

for some integer a. (The code is called narrow-sense if $a = 1$.)

Lemma 6.25

A BCH code defined in Definition 6.24 has $d \geq \delta$.

δ is referred to the designed distance.

Distance of BCH Codes: Proof of Lemma 6.25

Let α be the primitive element in $\mathbb{F}_{q^m}$. By construction, $\alpha^a, \cdots, \alpha^{a+\delta-2}$ are roots of the generator polynomial $g(x)$.
That is, $\forall \boldsymbol{c} \in \mathcal{C}$, the generating function $c(x)$ satisfies $c\left(\alpha^i\right) = 0$, $a \le i \le a+\delta-2$. In matrix format,

$$\begin{bmatrix} 1 & \alpha^a & \cdots & \alpha^{a(n-1)} \\ 1 & \alpha^{a+1} & \cdots & \alpha^{(a+1)(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{a+\delta-2} & \cdots & \alpha^{(a+\delta-1)(n-1)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \mathbf{0} \tag{3}$$

Any $(\delta-1)$-column submatrix is a Vandermonde matrix and hence of full rank. This implies $d \ge \delta$. $\Diamond$

Remark: The matrix in (3) is in $\mathbb{F}_{q^m}^{(\delta-1)\times n}$ while the vector $\boldsymbol{c} \in \mathcal{C} \subset \mathbb{F}_q^n$. Hence the matrix is not a parity-check matrix when $m > 1$.

Example: Reed-Solomon Codes

Recall that a RS-code $\mathcal{C}[n, k, n-k+1]$ is built on $\mathbb{F}_q$ with typically $n = q - 1$.

Compare the parity-check matrix of a RS-code (Theorem 6.5) and Equation (3). It is clear that a RS-code is a special case of a BCH code with $m = 1$.

In particular, suppose that we are asked to build a BCH code over $\mathbb{F}_q$ with $n = q - 1$ and $d \geq \delta = n - k + 1$. We find $M^{(i)}(x) \subset \mathbb{F}_q[x]$, $1 \leq i \leq 1 + \delta - 2 = n - k$. Since $M^{(i)}(x) \subset \mathbb{F}_q[x]$, it follows that $M^{(i)}(x) = x - \alpha^i$. Hence $g(x) = \prod_{i=1}^{n-k}\left(x - \alpha^i\right)$ and the generator matrix can be constructed (in a different form of that in Theorem 6.5) and good for implementation). Its parity-check matrix is given by the matrix in Equation (3). RS decoder can be directly applied for decoding.

Example: Binary BCH Codes

We have learned binary Hamming codes. The distance is always 3. The question is how to construct a binary code with large distance.

For example, how to construct a binary code of length 15 and $d \geq 5$?

1. For binary codes, use $\mathbb{F}_2$. $n = 15 = 2^4 - 1$ hence $m = 4$.
2. $\delta = 5$ implies $g(x) = \mathrm{lcm}\left(M^{(1)}(x), M^{(2)}(x), M^{(3)}(x), M^{(4)}(x)\right)$.
3. The relevant cyclotomic cosets of 2 modulo 15 include $\mathcal{C}_1 = \{1, 2, 4, 8\}$ and $\mathcal{C}_3 = \{3, 6, 9, 12\}$. Hence $M^{(1)}(x) = \prod_{i \in \mathcal{C}_1} \left(x - \alpha^i\right) = M^{(2)}(x) = M^{(4)}(x)$ and $M^{(3)}(x) = \prod_{i \in \mathcal{C}_3} \left(x - \alpha^i\right)$. Furthermore,
$$g(x) = M^{(1)}(x) \cdot M^{(3)}(x).$$
4. Find the generator matrix and parity-check matrix according to Theorems 6.20 and 6.22 respectively.

From Hamming to BCH

Example: A binary code of length 15 and $d \geq 5$?

$$\begin{aligned} g(x) &= \operatorname{lcm}\left(M^{(1)}(x), M^{(2)}(x), M^{(3)}(x), M^{(4)}(x)\right) \\ &= \operatorname{lcm}\left(M^{(1)}(x), M^{(3)}(x)\right) \\ &= M^{(1)}(x) \times M^{(3)}(x). \end{aligned}$$

RS codes are special cases of BCH codes ($m = 1$).

Have learned $[7, 4, 3]$ Hamming code.

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

Another view:

Let α be a primitive element of $\mathbb{F}_8$ that satisfies $\alpha^3 = \alpha + 1$.
The parity check matrix can be written as

$$\begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \end{bmatrix}.$$

(Column vectors in $\mathbb{F}_2^3 \mapsto$ Elements in $\mathbb{F}_8$)

From Hamming to BCH: Larger Distance

Binary BCH codes with $d \geq 5$:

$$\begin{aligned} g(x) &= \operatorname{lcm}\left(M^{(1)}(x), M^{(2)}(x), M^{(3)}(x), M^{(4)}(x)\right) \\ &= \operatorname{lcm}\left(M^{(1)}(x), M^{(3)}(x)\right). \end{aligned}$$

It holds that

$$\begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \\ 1 & \alpha^4 & \alpha & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \end{bmatrix} \boldsymbol{c} = \mathbf{0}$$

But $c(\alpha) = 0 \Rightarrow \begin{cases} c\left(\alpha^2\right) = c(\alpha)^2 = 0 \\ c\left(\alpha^4\right) = c(\alpha)^4 = 0 \end{cases}$

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

Eventually, we get a $[7, 1, 7]$ code $\mathcal{C} = \{0000000,\ 1111111\}$.