CS 860 Topics in Coding Theory		Sept. 4, 2025
	Lecture 1	
Lecturer: Elena Grigorescu		*Scribe: Tiger Wu*

1 What is Error Correcting Code (ECC)

Communications and storage suffer from data corruption, which may occur randomly [Sha48] or through an adversarial process [Ham50].

For communication we can consider a sender $\mathcal{S}$ sending a word $m \in \Sigma^k$ to a receiver $\mathcal{R}$:

$$\mathcal{S} \xrightarrow{\text{Encode}} \boxed{\mathsf{Enc}(m)} \xrightarrow{\text{Noisy channel}} \boxed{\mathsf{Enc}(m) + \text{noise}} \xrightarrow{\text{Decode}} \mathcal{R}$$

The general goal is to come up with methods of redundancy in such a way that $\mathcal{R}$ can map $\boxed{\mathsf{Enc}(m) + \text{noise}}$ back to m.

2

Definition 1 (Error-Correcting Code (ECC)) *ECC is a pair of maps* $(\mathsf{Enc}, \mathsf{Dec})$ *consisting of an injective map* $\mathsf{Enc} : \Sigma^k \to \Sigma^n$ *and* $\mathsf{Dec} : \Sigma^n \to \Sigma^k$.

We refer to Σ^k as the message space and $m \in \Sigma^k$ as a message. We refer to $C = \mathsf{Enc}(\Sigma^k) \subseteq \Sigma^n$ as the set of codewords and $c \in C$ as a codeword. k is the message length and n is the code length. Sometimes, we don't care about the messages and focus only on codewords C. Another classical parameter is the minimum distance d. The intuition is that if we make code sparse in the code space, then from the neighbors of a code, we can map to the original code. We elaborate on what is meant by distance below. We will mostly deal with Hamming distance.

Definition 2 (Hamming Distance) *For* $x, y \in \Sigma^n$*, the hamming distance, denoted by* $\Delta(x, y)$ *is the following*

$$\Delta(x, y) = |\{i \mid x_i \neq y_i\}|$$

where x_i *refers to the* i*th bit of* x*, similarly with* y_i *and* y.

Definition 3 (Distance of code) *For code* $C \subseteq \Sigma^n$*, the minimum distance of* C*, denotes as* $\Delta(C)$ *is*

$$\Delta(c) = \min_{\substack{x,y \in C \\ x \neq y}} \Delta(x, y)$$

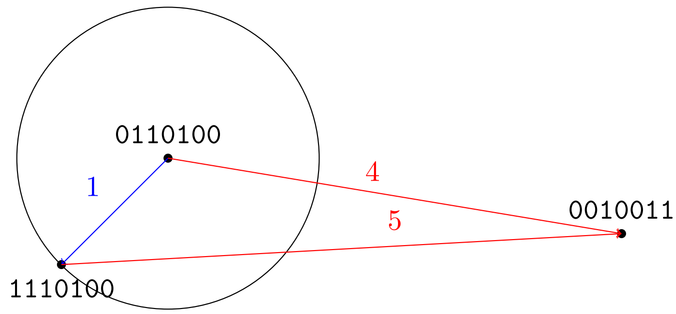
Definition 4 (Relative Minimial Distance) *The relative minimial distance of* $C \in \Sigma^n$ *is*

$$\frac{\Delta(C)}{n}.$$

Example

Consider the code 0001 $\in \{0,1\}^4$. The codes '1001', '0101', '0011', and '0000' are distance 1 away from 0001. If we let $C \subseteq \{0,1\}^4$ to be codes where $\Delta(C) = 3$, Then when one error occurs to '0001' $\in C$, i.e., '0001' $\to$ '1001', we can map it back to '0001'.

Continuing with the above idea, let $C \subseteq \{0,1\}^7$ and $\Delta(C) = 4$. Let '0110100' and '0010011' $\in C$. We can see in the figure below that when 1 error occurs when transmitting '0110100', (i.e., 1110100), we can map it to '0110100' without confusing it with '0010011'



Remark When the number of errors, e, is less than $\Delta(C)/2$, there is a unique code within e errors from the recovered word. In this case, we say that C is e error correcting.

3 Types of Noise

There are three types of noise:

1. **Erasure:**
   - Some symbols in a word are erased
   - Location of erasure is known
   - Example: 00111 $\to$ 0_1_1
2. **Errors/Substitution/Bit-flips**
   - Location of error is now known
   - Example: 00111 $\to$ 01001
3. **Deletion/Insertion**
   - Location unknown
   - Example: 00111 $\to$ 011
00110 $\to$ 0011011

There are also two models of how noise occurs.

1. **Stochastic (Shannon's model)**. Where errors occur randomly.
2. **Adversarial (Hamming's model)**. Where the worst-case errors occur.

3.1 Basic Examples

3.1.1 Correcting from 1-erasure

Let $m \in \{0,1\}^k$.

Method 1. Duplication: For $(m_0, m_1, \ldots, m_k)$ we can duplicate each symbol and send the code

$$(m_0, m_0, m_1, m_1, \ldots, m_k, m_k).$$

We denote this code as C_{dup}. In this case, if c_i is lost, we can find c_i by taking c_{i+1} if i is odd and c_{i-1} if i is even.

Methods 2. parity bits: For $(m_0, m_1, \ldots, m_k)$ we add a parity bit at the end,

$$(m_1, \ldots, m_k, \sum_{j=1}^{k} m_j).$$

We denote this code as C_{par}. In this case, if c_i is lost, we can find c_i by taking the parity of

$$\left(\sum_{j=1}^{i-1} c_j\right) + \left(\sum_{j=i+1}^{k} c_j\right) + c_{k+1}.$$

Which of the above codes is better?

We can observe that $\Delta(C_{\text{dup}}) = 2$ and $\Delta(C_{\text{par}}) = 2$, therefore both have $O(n)$ encoders and decoders. However, these codes have different rates.

Definition 5 *The rate of code $C \in \Sigma^n$ is*

$$R(C) = \frac{\log |C|}{n \log |\Sigma|}.$$

With respect to messages $m \in \Sigma^k$, the rate is

$$R(C) = \frac{k}{n}.$$

Ideally, we want the rate of a code to be close to 1. For the code above, we can compute

$$R(C_{\text{dup}}) = \frac{1}{2} \text{ and } R(C_{\text{par}}) = \frac{n-1}{n}.$$

3.1.2 Correcting from 1-error (bit-flip)

In this scenario, the duplication code from above does not work. For c_{2x+1} and c_{2x+2}, there is no way of telling which bit is flipped. So instead of duplicating each bit twice, we can duplicate each bit three times.

Method 1. Duplication: For $(m_0, \ldots, m_n)$ we can use the code

$$(m_0, m_0, m_0, \ldots, m_k, m_k, m_k).$$

We denote this code as C_{triple}. The rate of this code is

$$R(C_{\text{triple}}) = \frac{1}{3}.$$

Method 2. Varshamov-Tenengolts: Send

$$(c_0, c_1, \ldots, c_n) \in \{0,1\}^n$$

such that

$$c_1 + 2c_2 + 3c_3 + \ldots + nc_n \equiv 0 \bmod 2n+1.$$

We denote this code as C_{vt}

Claim 6 *C_{vt} is 1-error correcting.*

Proof Consider $c \in C_{\text{vt}} \subseteq \{0,1\}^n$. If the ith bit of c is flipped to $c_i' = 1 - c_i$. Then

$$c_1 + 2c_2 + \ldots + i(1-c_i) + \ldots + nc_n = s$$

$$c_1 + 2c_2 + \ldots + i(1-c_i) + \ldots + nc_n - \left(\sum_{i=1}^{n} ic_i\right) \equiv 0 \bmod (2n-1)$$

$$i(1-2c_i) \equiv -s \bmod (2n-1)$$

Since $1 \leq i \leq n$ and $c_i \in \{0,1\}$, i is unique. ■

Later in the class, we show that C_{vt} can correct even 1 deletion [Lcv66].

4 Distance vs. Detection/Correction

Definition 7 (t-error correcting) *$C \subseteq \Sigma^n$ is t-error correcting if $\forall x \in C$, $\forall y \in \Sigma^n$ such that $\Delta(x,y) \leq t$, x is the unique element of C such that $\Delta(x,y) \leq t$.*

Definition 8 (t-erasure correcting) *$C \subseteq \Sigma^n$ is t-erasure correcting if $\forall x \in C$ $\forall y \in \Sigma^n \cup \{?\}$ such that $\Delta(x,y) \leq t$, then x is the unique element of C such that $\Delta(x,y) \leq t$.*

Definition 9 (t-error detecting) *$C \subseteq \Sigma^n$ is $\forall x \in C$ $\forall y \in \Sigma^n$ such that $\Delta(x,y) \leq t$, then $y \in C$ if and only if $y = x$.*

4.1 Homework

Let $C \subseteq \Sigma^n$.

1. $\Delta(C) \geq 2t+1$ if and only if C is t-error correcting.
2. $\Delta(C) \geq 2t+1$ if and only if C is $2t$-error detecting.
3. $\Delta(C) \geq 2t+1$ if and only if C is $2t$-erasure corrrecting.

Observe the tension between $|C|$ and the error correcting potential (i.e., R vs. Δ).

4.2 Big Meta Questions

Question 1 *For a given d, what is the max $|C|$, $C \subseteq \Sigma^n$ such that $\Delta(C) = d$?*

Question 2 *How to construct codes achieving max tradeoffs?*

Question 3 *How to design codes with good/optimal rate/distance tradeoff which are also efficiently encodable and decodable?*

5 Hamming Bounds

Definition 10 (Hamming ball/set) *For $x \in \Sigma^n$ and $r \in \mathbb{Z}^{\geq 0}$, hamming ball/set $B(x, r)$ is*

$$B(x, y) = \{y \in \Sigma^n \mid \Delta(x, y) \leq r\}.$$

We write $B_n(r)$ to denote $B_n(x, r)$. For $\Sigma = \{0, 1\}$, it can be observed that $|B_n(1)| = n + 1$ and $|B_n(2)| = \Theta(n^2)$.

Claim 11 *$C \in \Sigma^n$ has distance (odd) d if and only if*

$$B(x, \frac{d-1}{2}) \cap B(y, \frac{d-1}{2}) = \emptyset$$

for all $x \neq y \in C$.

Theorem 12 (Hamming bound) *If $C \subseteq \Sigma^n$ has distance d, then*

$$|C| \leq \frac{|\Sigma^n|}{B_n(\lfloor \frac{d-1}{2} \rfloor)}.$$

Proof For $y \in \Sigma^n$, y is in at most 1 ball. ■

Hamming codes perfectly pack the code space. It is also called the perfect codes.

5.1 Hamming Code

Hamming Code $\text{Ham}_{n,k,d}$ is parameterized by $n = 2^t - 1$, $k = 2^t - 1 - t$ and $d = 3$. Hamming code $\text{Ham}_{n,k,d} \subseteq \mathbb{F}_2^n$ is the set

$$\left\{ c \in \mathbb{F}_2^n \mid H \cdot c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\}$$

where $H \in \mathbb{F}_2^{(n-k)\times n}$ and the right most matrix has t rows. In general H has dimension $t \times (2^t - 1)$.

For example if $n = 7$ and $t = 3$,

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

Then every $c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_7 \end{bmatrix}$ where $Hc = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ is going to be a code word.

Theorem 13 *The code* $Ham_{n,k,d}$ *has*

1. $|Ham| = \dfrac{2^n}{n+1}$,
2. $d = 3$.

6 Puzzle

There are n students in this classroom who all want to get an A. The professor promises they all get A's if they win as a team the following game:

- You (each student) has a (randomly) black or white sticker on your forehead. You cannot see your own sticker, but you can see others' stickers.
- You (the students) act as a team; all get A's or all fail.
- You are not allowed to communicate after the stickers were placed on your head. (But can plan a strategy beforehand).
- All students answer simultaneously.
- You can choose to guess the sticker's color on your forehead or pass.
- The team wins if at least one player guessed a color and all who guessed a color guessed correctly.

What is the maximum probability of winning, and what is the strategy?

References

[Ham50] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[Lcv66] VI Lcvenshtcin. Binary coors capable or 'correcting deletions, insertions, and reversals. In *Soviet physics-doklady*, volume 10, 1966.

[Sha48] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.