

A Quick Introduction to Classical Error-Correcting Codes

Navin Kashyap

Department of Electrical Communication Engineering
Indian Institute of Science

Outline of Talk

Background and Motivation

Linear Codes

Encoding and Decoding

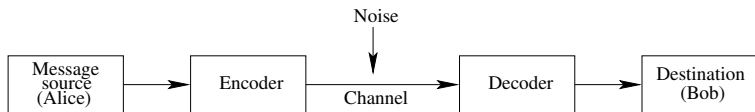
Bounds on the Parameters of Linear Codes

Some Code Constructions

Some Code Families

Applications of Error-Correcting Codes

The Communications System Model



Requirements for information transmission:

- ▶ Cost of transmission should be low
- ▶ Information should be transmitted reliably in the presence of channel noise
- ▶ Information should be transmitted securely in the presence of an adversary/eavesdropper

What is Coding?

Coding is the representation of information using symbols, often 0's and 1's.

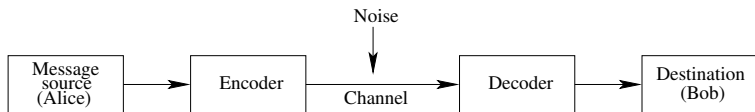
Comes in three flavours —

Source Coding: Enables Alice to *compress* her message to save on cost of transmission.

Channel Coding: Enables Alice to *send her message reliably* to Bob, by introducing some mechanism to counter channel noise.

Secrecy Coding (aka **Cryptography**): Enables Alice to *encrypt* her message to foil her enemies.

Channel Noise



Channel noise follows a model known to both Alice and Bob, who intend to fully make use of this knowledge to encode and decode.

Noise may be deterministic or random; if random, it follows a probabilistic model.

The effect of noise is to introduce errors in the transmitted message.

The goal of channel coding is to reduce or eliminate the effects of channel noise.

Three Types of Channel Codes

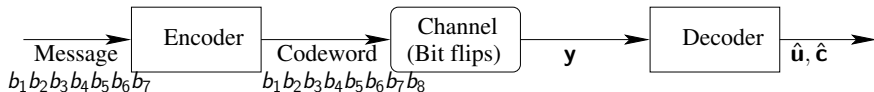
Error-Detecting Codes: Allows Bob to *detect* errors in received message; useful in random noise situations.

Error-Correcting Codes: Allows Bob to *correct* errors in received message; useful in random noise situations.

Constrained Codes: Alice encodes the message in such a way as to *prevent* errors from corrupting the message; useful in deterministic noise situations.

All coding schemes work by adding **redundancy** to the message to compensate for errors: more symbols are transmitted than are in the original message.

A Simple Error-Detecting Code



Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

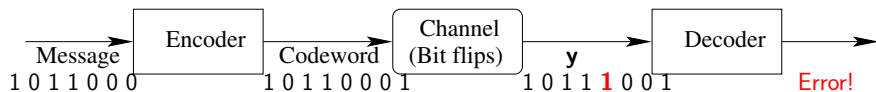
Message: 7-bit binary sequence ($b_1, b_2, b_3, b_4, b_5, b_6, b_7$)

Encoder: Adds 8th bit, called a **parity bit**, b_8 , so that (b_1, b_2, \dots, b_8) contains an even number of 1s:

$$\sum_{i=1}^8 b_i \equiv 0 \pmod{2}$$

Rate: The number of message bits per coded bit is $\frac{7}{8}$.

A Simple Error-Detecting Code



Decoder:

$$\hat{\mathbf{c}} = \begin{cases} \mathbf{y} & \text{if } \mathbf{y} \text{ has even parity} \\ \text{ERROR} & \text{if } \mathbf{y} \text{ has odd parity} \end{cases}$$

This code detects an odd number of errors.

Applications of Error-Detecting Codes

Computer network communication protocols:

Network protocols such as TCP/IP use error-detecting codes extensively to determine if data packets sent across a network have been corrupted or not.

Usual remedy for receipt of a corrupted data packet is to request **retransmission**.

Applications of Error-Detecting Codes

Computer network communication protocols:

Network protocols such as TCP/IP use error-detecting codes extensively to determine if data packets sent across a network have been corrupted or not.

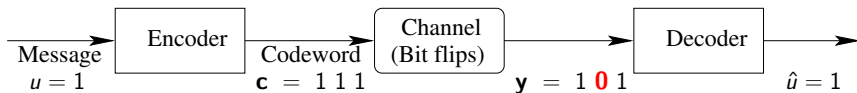
Usual remedy for receipt of a corrupted data packet is to request **retransmission**.

International Standard Book Numbers (ISBN)

- ▶ ISBN-10 numbers have 10 digits, and are of the form x-xxx-xxxxx-x.
- ▶ The first nine digits in an ISBN record information about the book: country, publisher, title and edition;
- ▶ The tenth digit is a **check digit**.



A Simple Error-Correcting Code

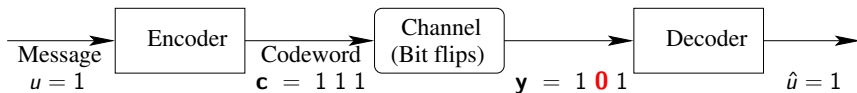


Channel Noise: Flips bits at random ($0 \rightarrow 1$, $1 \rightarrow 0$).

Message: Single bit, $b = 0$ or 1 .

Encoder: $0 \mapsto 000$, $1 \mapsto 111$. Thus, **rate** = $\frac{1}{3}$.

A Simple Error-Correcting Code



Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

Message: Single bit, $b = 0$ or 1 .

Encoder: $0 \mapsto 000, 1 \mapsto 111$. Thus, **rate** $= \frac{1}{3}$.

Decoder: Majority rule — if more 0s than 1s received, then decode as 0; else decode as 1.

This code can correct single errors.

The Length-7 Hamming Code

Discovered by Richard Hamming (1950)

- Encodes a 4-bit message into a 7-bit codeword:

$$\underbrace{u_1 \ u_2 \ u_3 \ u_4}_{\text{information bits}} \longmapsto u_1 \ u_2 \ u_3 \ u_4 \ \underbrace{p_1 \ p_2 \ p_3}_{\text{parity bits}}$$

- Thus, $\text{rate} = \frac{4}{7}$.

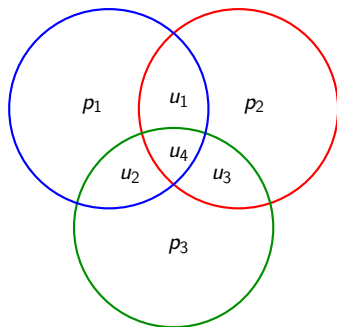
The Length-7 Hamming Code

Discovered by Richard Hamming (1950)

- Encodes a 4-bit message into a 7-bit codeword:

$$\underbrace{u_1 \ u_2 \ u_3 \ u_4}_{\text{information bits}} \longmapsto u_1 \ u_2 \ u_3 \ u_4 \underbrace{p_1 \ p_2 \ p_3}_{\text{parity bits}}$$

- Thus, $\text{rate} = \frac{4}{7}$.



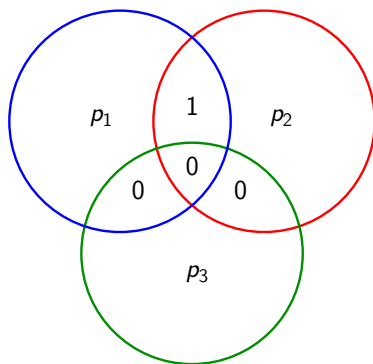
For each choice of (u_1, u_2, u_3, u_4) , there is a unique choice of (p_1, p_2, p_3) that makes each circle have an even number of 1s.

$$u_1 + u_2 + u_4 + p_1 \equiv 0 \pmod{2}$$

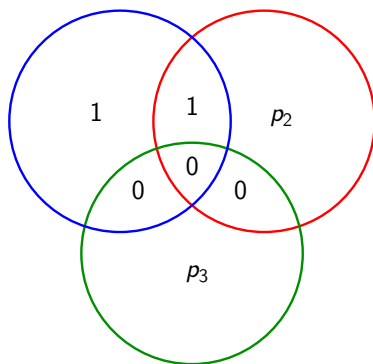
$$u_1 + u_3 + u_4 + p_2 \equiv 0 \pmod{2}$$

$$u_2 + u_3 + u_4 + p_3 \equiv 0 \pmod{2}$$

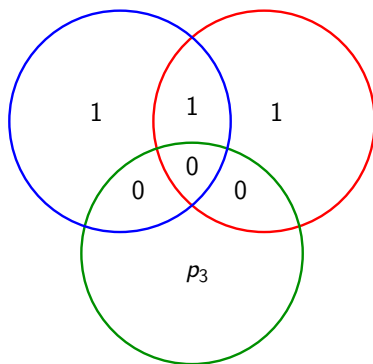
The Length-7 Hamming Code



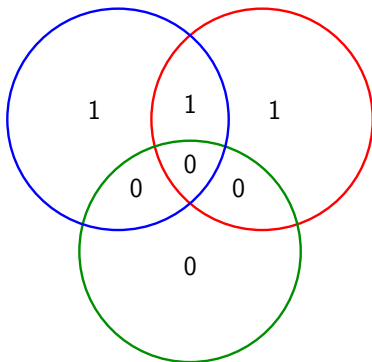
The Length-7 Hamming Code



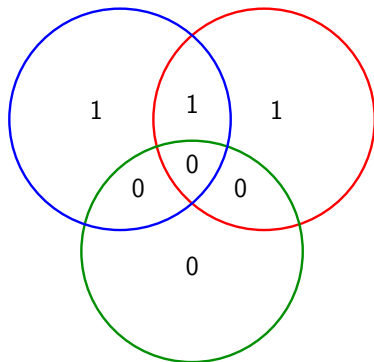
The Length-7 Hamming Code



The Length-7 Hamming Code



The Length-7 Hamming Code



$u_1 \ u_2 \ u_3 \ u_4 \ p_1 \ p_2 \ p_3 = 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0$
is a codeword of the Hamming code.

The Length-7 Hamming Code: Single-Error Correction

Encoder (Alice): $(u_1, u_2, u_3, u_4) \mapsto (u_1, u_2, u_3, u_4, p_1, p_2, p_3)$

Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

The Length-7 Hamming Code: Single-Error Correction

Encoder (Alice): $(u_1, u_2, u_3, u_4) \mapsto (u_1, u_2, u_3, u_4, p_1, p_2, p_3)$

Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

Suppose that Alice sends **1000110**, but Bob receives **1100110**:

$$1\ 0\ 0\ 0\ 1\ 1\ 0 \longrightarrow 1\ 1\ 0\ 0\ 1\ 1\ 0$$

The Length-7 Hamming Code: Single-Error Correction

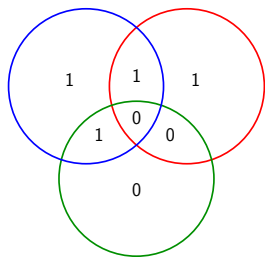
Encoder (Alice): $(u_1, u_2, u_3, u_4) \mapsto (u_1, u_2, u_3, u_4, p_1, p_2, p_3)$

Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

Suppose that Alice sends **1000110**, but Bob receives **1100110**:

$$1\ 0\ 0\ 0\ 1\ 1\ 0 \longrightarrow 1\ 1\ 0\ 0\ 1\ 1\ 0$$

Decoder (Bob):



1. Identify circles with wrong parity
2. Flip bit common to those circles only

The Length-7 Hamming Code: Single-Error Correction

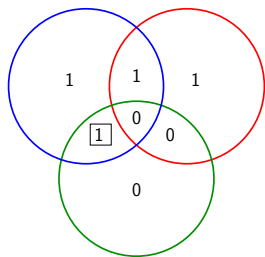
Encoder (Alice): $(u_1, u_2, u_3, u_4) \mapsto (u_1, u_2, u_3, u_4, p_1, p_2, p_3)$

Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

Suppose that Alice sends **1000110**, but Bob receives **1100110**:

$$1\ 0\ 0\ 0\ 1\ 1\ 0 \longrightarrow 1\ \mathbf{1}\ 0\ 0\ 1\ 1\ 0$$

Decoder (Bob):



1. Identify circles with wrong parity
2. Flip bit common to those circles only

The Length-7 Hamming Code: Single-Error Correction

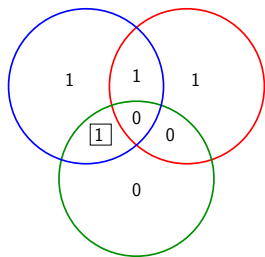
Encoder (Alice): $(u_1, u_2, u_3, u_4) \mapsto (u_1, u_2, u_3, u_4, p_1, p_2, p_3)$

Channel Noise: Flips bits at random ($0 \rightarrow 1, 1 \rightarrow 0$).

Suppose that Alice sends **1000110**, but Bob receives **1100110**:

$$1\ 0\ 0\ 0\ 1\ 1\ 0 \longrightarrow 1\ \mathbf{0}\ 0\ 0\ 1\ 1\ 0$$

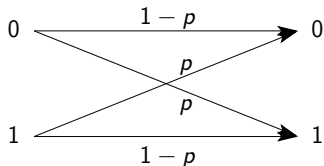
Decoder (Bob):



1. Identify circles with wrong parity
2. Flip bit common to those circles only

Single errors can be corrected.

Memoryless Binary Symmetric Channel



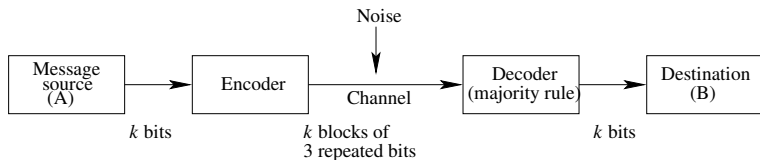
- ▶ Alice wants to send a k -bit message to Bob.
- ▶ The message is to be encoded and transmitted across a memoryless BSC.
- ▶ Each bit transmitted across the channel gets flipped with probability p , independently of other bits.
- ▶ For concreteness, $k = 10,000$ and $p = 0.001$.

Uncoded Transmission

- ▶ Rate = 1
- ▶ Probability that the entire k -bit message is recovered by Bob:

$$\begin{aligned} P_C &= (1 - p)^k \\ &= 0.000045 \quad \text{for } k = 10,000 \text{ and } p = 0.001 \end{aligned}$$

Repetition Codes



The probability that a single message bit is decoded correctly is

$$P_{b,C} = (1 - p)^3 + 3p(1 - p)^2$$

So, the probability that entire k -bit message is decoded correctly is

$$P_C = (P_{b,C})^k$$

$$\approx 0.97 \quad \text{for } k = 10,000, p = 0.001$$

(Note: 30,000 coded bits are actually transmitted)

Length-7 Hamming Code

To transmit a k -bit message, we first divide the message into 4-bit blocks, encode each using a 7-bit codeword, and transmit.

Thus, $k/4$ codewords (i.e., $7k/4$ coded bits) are transmitted.

Length-7 Hamming Code

To transmit a k -bit message, we first divide the message into 4-bit blocks, encode each using a 7-bit codeword, and transmit.

Thus, $k/4$ codewords (i.e., $7k/4$ coded bits) are transmitted.

- ▶ The prob. that a transmitted 7-bit codeword is decoded correctly is

$$\begin{aligned}\Pr[\text{at most one of the 7 bits is flipped}] \\ = (1 - p)^7 + 7p(1 - p)^6\end{aligned}$$

- ▶ Hence, the prob. that the entire k -bit message is recovered correctly by Bob is

$$\begin{aligned}P_C &= \left[(1 - p)^7 + 7p(1 - p)^6 \right]^{k/4} \\ &\approx 0.95 \quad \text{for } k = 10,000 \text{ and } p = 0.001\end{aligned}$$

Length-7 Hamming Code

To transmit a k -bit message, we first divide the message into 4-bit blocks, encode each using a 7-bit codeword, and transmit.

Thus, $k/4$ codewords (i.e., $7k/4$ coded bits) are transmitted.

- ▶ The prob. that a transmitted 7-bit codeword is decoded correctly is

$$\begin{aligned}\Pr[\text{at most one of the 7 bits is flipped}] \\ = (1 - p)^7 + 7p(1 - p)^6\end{aligned}$$

- ▶ Hence, the prob. that the entire k -bit message is recovered correctly by Bob is

$$\begin{aligned}P_C &= \left[(1 - p)^7 + 7p(1 - p)^6 \right]^{k/4} \\ &\approx 0.95 \quad \text{for } k = 10,000 \text{ and } p = 0.001\end{aligned}$$

- ▶ $(7 \times 10000)/4 = 17,500$ coded bits are transmitted.

Shannon's Noisy Channel Coding Theorem

Channel Model: BSC with crossover probability p

BSC Channel Capacity: $C(p) = 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$.

Theorem [Claude Shannon (1948)]:

If $R < C(p)$ and k is sufficiently large, there exists a code which can encode k message bits into $n = k/R$ coded bits to be transmitted across the channel, such that a decoder at the channel output can recover all k message bits correctly with probability close to 1.

Shannon's Noisy Channel Coding Theorem

Channel Model: BSC with crossover probability p

BSC Channel Capacity: $C(p) = 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$.

Theorem [Claude Shannon (1948)]:

If $R < C(p)$ and k is sufficiently large, there exists a code which can encode k message bits into $n = k/R$ coded bits to be transmitted across the channel, such that a decoder at the channel output can recover all k message bits correctly with probability close to 1.

► $C(0.001) \approx 0.9886$.

Shannon's Noisy Channel Coding Theorem

Channel Model: BSC with crossover probability p

BSC Channel Capacity: $C(p) = 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$.

Theorem [Claude Shannon (1948)]:

If $R < C(p)$ and k is sufficiently large, there exists a code which can encode k message bits into $n = k/R$ coded bits to be transmitted across the channel, such that a decoder at the channel output can recover all k message bits correctly with probability close to 1.

- ▶ $C(0.001) \approx 0.9886$.
- ▶ This means that, to ensure that a 10,000-bit message can be recovered correctly with prob. close to 1, it should be enough to transmit about $10000/0.9886 \approx 10,150$ coded bits!

The Goal of Coding Theory

The goal of coding theory is to design coding schemes that can approach Shannon's performance guarantees, while still being relatively easy to implement in practice.

Low complexity, good error-correction capability

Linear Codes

Let \mathbb{F}_q denote a finite field of size q . For concreteness, take $q = 2$, i.e., \mathbb{F}_2 is the binary field $\{0, 1\}$ with modulo-2 arithmetic.

Notation: $\mathbb{F}_q^n = \{(x_1, x_2, \dots, x_n) : x_i \in \mathbb{F}_q\}$.

Clearly, $|\mathbb{F}_q^n| = |\mathbb{F}_q|^n = q^n$.

Definition

A **linear code** \mathcal{C} of **blocklength** n over \mathbb{F}_q is a linear subspace of \mathbb{F}_q^n :

- ▶ for each pair of codewords $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$, and for any $\alpha_1, \alpha_2 \in \mathbb{F}$, we also have $\alpha_1 \mathbf{c}_1 + \alpha_2 \mathbf{c}_2 \in \mathcal{C}$.

The **dimension** of a code \mathcal{C} is the dimension of \mathcal{C} as a linear subspace of \mathbb{F}_q^n over \mathbb{F}_q ; denoted by $\dim(\mathcal{C})$ or $\dim_{\mathbb{F}_q}(\mathcal{C})$.

An **$[n, k]$ linear code** is a linear code of blocklength n and dim. k .

Rate

Proposition: An $[n, k]$ linear code \mathcal{C} over \mathbb{F}_q has q^k codewords.

Proof: Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ be a basis of the subspace \mathcal{C} .

- ▶ Every codeword (i.e., vector) $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination

$$\mathbf{c} = \alpha_1 \cdot \mathbf{c}_1 + \alpha_2 \cdot \mathbf{c}_2 + \dots + \alpha_k \cdot \mathbf{c}_k, \quad \text{with } \alpha_j \in \mathbb{F}_q \text{ for all } j$$

- ▶ Thus, there is a 1-1 correspondence between codewords $\mathbf{c} \in \mathcal{C}$ and k -tuples of coefficients $(\alpha_1, \dots, \alpha_k) \in \mathbb{F}_q^k$.
- ▶ Hence, $|\mathcal{C}| = |\mathbb{F}_q^k| = q^k$. □

Rate

Proposition: An $[n, k]$ linear code \mathcal{C} over \mathbb{F}_q has q^k codewords.

Proof: Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ be a basis of the subspace \mathcal{C} .

- ▶ Every codeword (i.e., vector) $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination

$$\mathbf{c} = \alpha_1 \cdot \mathbf{c}_1 + \alpha_2 \cdot \mathbf{c}_2 + \dots + \alpha_k \cdot \mathbf{c}_k, \quad \text{with } \alpha_j \in \mathbb{F}_q \text{ for all } j$$

- ▶ Thus, there is a 1-1 correspondence between codewords $\mathbf{c} \in \mathcal{C}$ and k -tuples of coefficients $(\alpha_1, \dots, \alpha_k) \in \mathbb{F}_q^k$.
- ▶ Hence, $|\mathcal{C}| = |\mathbb{F}_q^k| = q^k$. □

The **rate** of an $[n, k]$ linear code over \mathbb{F}_q is

$$R = \frac{k}{n}.$$

Examples

- ▶ The **repetition code** over \mathbb{F}_2 : $\{\underbrace{00 \dots 0}_n, \underbrace{11 \dots 1}_n\}$

This is a linear code over \mathbb{F}_2 with blocklength n and dim. 1.
In other words, this is an $[n, 1]$ binary linear code.

Examples

- ▶ The **repetition code** over \mathbb{F}_2 : $\{\underbrace{00 \dots 0}_n, \underbrace{11 \dots 1}_n\}$
 n 0s n 1s

This is a linear code over \mathbb{F}_2 with blocklength n and dim. 1.
In other words, this is an $[n, 1]$ binary linear code.

- ▶ The **single parity-check code** over \mathbb{F}_2 :

$$\begin{aligned}\mathcal{C} &= \{x_1 x_2 \dots x_n : x_1 + x_2 + \dots + x_n \equiv 0 \pmod{2}\} \\ &= \text{nullspace}(H), \quad \text{where } H = [\underbrace{1 \ 1 \ \dots \ 1}_n].\end{aligned}$$

n columns

Examples

- ▶ The **repetition code** over \mathbb{F}_2 : $\{\underbrace{00 \dots 0}_n, \underbrace{11 \dots 1}_n\}$
 n 0s n 1s

This is a linear code over \mathbb{F}_2 with blocklength n and dim. 1.
In other words, this is an $[n, 1]$ binary linear code.

- ▶ The **single parity-check code** over \mathbb{F}_2 :

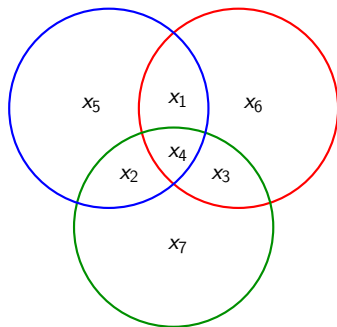
$$\begin{aligned}\mathcal{C} &= \{x_1 x_2 \dots x_n : x_1 + x_2 + \dots + x_n \equiv 0 \pmod{2}\} \\ &= \text{nullspace}(H), \quad \text{where } H = \underbrace{[1 \ 1 \ \dots \ 1]}_{n \text{ columns}}.\end{aligned}$$

By the **rank-nullity theorem**,

$$\dim(\mathcal{C}) = n - \text{rank}(H) = n - 1.$$

Thus, \mathcal{C} is an $[n, n - 1]$ binary linear code.

Example: The Length-7 Hamming Code



A binary word

$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$

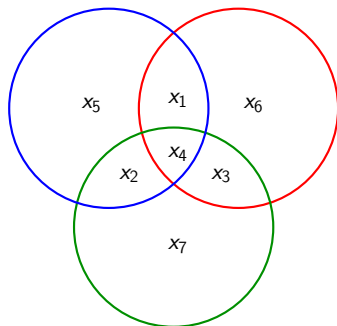
is in the Hamming code iff

$$x_1 + x_2 + x_4 + x_5 \equiv 0 \pmod{2}$$

$$x_1 + x_3 + x_4 + x_6 \equiv 0 \pmod{2}$$

$$x_2 + x_3 + x_4 + x_7 \equiv 0 \pmod{2}$$

Example: The Length-7 Hamming Code



A binary word

$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$

is in the Hamming code iff

$$x_1 + x_2 + x_4 + x_5 \equiv 0 \pmod{2}$$

$$x_1 + x_3 + x_4 + x_6 \equiv 0 \pmod{2}$$

$$x_2 + x_3 + x_4 + x_7 \equiv 0 \pmod{2}$$

Re-write these equations in matrix form (over \mathbb{F}_2) as

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Example: The Length-7 Hamming Code

Thus, a binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code \mathcal{C} iff

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_H \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}}_{\mathbf{x}^T} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}.$$

In other words, the Hamming code \mathcal{C} is equal to $\text{nullspace}_{\mathbb{F}_2}(H)$.

Consequently, by the rank-nullity theorem,

- ▶ $\dim(\mathcal{C}) = n - \text{rank}_{\mathbb{F}_2}(H) = 7 - 3 = 4$.
- ▶ \mathcal{C} is a $[7, 4]$ binary linear code.

Minimum distance

Let $\mathbf{c} = (c_1, c_2, \dots, c_n)$ and $\mathbf{c}' \in (c'_1, c'_2, \dots, c'_n)$ be words in \mathbb{F}^n .

- ▶ The **Hamming distance** between \mathbf{c} and \mathbf{c}' is defined to be

$$d(\mathbf{c}, \mathbf{c}') = |\{i : c_i \neq c'_i\}|$$

- ▶ The **Hamming weight** of \mathbf{c} is defined to be

$$w(\mathbf{c}) = d(\mathbf{c}, \mathbf{0}) = |\{i : c_i \neq 0\}|$$

Definition

The **minimum distance** of a linear code \mathcal{C} is

$$\begin{aligned} d_{\min}(\mathcal{C}) &= \min_{\mathbf{c}, \mathbf{c}' \in \mathcal{C} : \mathbf{c} \neq \mathbf{c}'} d(\mathbf{c}, \mathbf{c}') \\ &= \min_{\mathbf{c} \in \mathcal{C} : \mathbf{c} \neq \mathbf{0}} w(\mathbf{c}) \end{aligned}$$

- ▶ An $[n, k, d]$ **linear code** is an $[n, k]$ linear code with min dist d .

Examples

- ▶ The **repetition code** over \mathbb{F}_2 : $\{\underbrace{00 \dots 0}_{n \text{ 0s}}, \underbrace{11 \dots 1}_{n \text{ 1s}}\}$

This is an $[n, 1, n]$ binary linear code.

Examples

- ▶ The **repetition code** over \mathbb{F}_2 : $\{\underbrace{00 \dots 0}_n, \underbrace{11 \dots 1}_n\}$

This is an $[n, 1, n]$ binary linear code.

- ▶ The **single parity-check code** over \mathbb{F}_2 :

$$\mathcal{C} = \{x_1 x_2 \dots x_n : x_1 + x_2 + \dots + x_n \equiv 0 \pmod{2}\}$$

Since there are no codewords of (odd) weight 1,
and all binary words of (even) weight 2 are in the code,

$$d_{\min}(\mathcal{C}) = 2.$$

Thus, this is an $[n, n-1, 2]$ binary linear code.

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code \mathcal{C} iff

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_H \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}}_{\mathbf{x}^T} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}.$$

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code \mathcal{C} iff

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code \mathcal{C} iff

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- ▶ \mathcal{C} has **no** codewords of weight 1, as no column of H is **0**.
- ▶ \mathcal{C} has **no** codewords of weight 2, as no two columns of H are identical.
- ▶ \mathcal{C} does have codewords of weight 3: e.g., the first three columns of H sum to **0** over \mathbb{F}_2 , so 1110000 is in \mathcal{C} .

Hence, $d_{\min}(\mathcal{C}) = 3$.

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code \mathcal{C} iff

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- ▶ \mathcal{C} has **no** codewords of weight 1, as no column of H is **0**.
- ▶ \mathcal{C} has **no** codewords of weight 2, as no two columns of H are identical.
- ▶ \mathcal{C} does have codewords of weight 3: e.g., the first three columns of H sum to **0** over \mathbb{F}_2 , so 1110000 is in \mathcal{C} .

Hence, $d_{\min}(\mathcal{C}) = 3$.

Thus, the Hamming code is a $[7, 4, 3]$ binary linear code.

Minimum Distance of $\mathcal{C} = \text{nullspace}(H)$

Let \mathcal{C} be a linear code of \mathbb{F} with parity-check matrix H , i.e., $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$.

Theorem: The minimum distance of \mathcal{C} is equal to the smallest number of columns of H that are linearly dependent over \mathbb{F} .

(Equivalently, $d_{\min}(\mathcal{C})$ is the largest integer d such that every collection of $d - 1$ columns of H is linearly independent over \mathbb{F} .)

Minimum Distance of $\mathcal{C} = \text{nullspace}(H)$

Let \mathcal{C} be a linear code of \mathbb{F} with parity-check matrix H , i.e., $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$.

Theorem: The minimum distance of \mathcal{C} is equal to the smallest number of columns of H that are linearly dependent over \mathbb{F} .

(Equivalently, $d_{\min}(\mathcal{C})$ is the largest integer d such that every collection of $d - 1$ columns of H is linearly independent over \mathbb{F} .)

If $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$, then H is called a **parity-check matrix** for \mathcal{C} .

Error Correction

Given: a length- n code \mathcal{C} .

An **error** is an event of changing an entry in a codeword.

A codeword $\mathbf{c} \in \mathcal{C}$ is transmitted, and $\mathbf{y} \in \mathbb{F}^n$ is received.
The number of errors that have occurred is $d(\mathbf{y}, \mathbf{c})$.

Definition

A code \mathcal{C} is **t -error-correcting** if there exists a decoding map $\mathcal{D} : \mathbb{F}^n \rightarrow \mathcal{C}$ such that
whenever $d(\mathbf{y}, \mathbf{c}) \leq t$, we have $\mathcal{D}(\mathbf{y}) = \mathbf{c}$.

Minimum Distance and Error Correction

Proposition

An $[n, k, d]$ code is t -error-correcting for $t < d/2$.

Proof: A codeword $\mathbf{c} \in \mathcal{C}$ is transmitted, and $\mathbf{y} \in \mathbb{F}^n$ is received. The number of errors that have occurred is $d(\mathbf{y}, \mathbf{c})$.

Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c}' \in \mathcal{C}} d(\mathbf{y}, \mathbf{c}')$.

If $d(\mathbf{y}, \mathbf{c}) < d/2$, then $\hat{\mathbf{c}} = \mathbf{c}$.



Erasures

- ▶ An **erasure** is an error whose location is known.
- ▶ Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \dots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \dots c_n$$

Erasures

- ▶ An **erasure** is an error whose location is known.
- ▶ Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \dots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \dots c_n$$

Proposition:

Let \mathcal{C} be an $[n, k, d]$ code over \mathbb{F} . There is a decoder for \mathcal{C} that corrects any occurrence of up to $d - 1$ erasures.

Erasures

- ▶ An **erasure** is an error whose location is known.
- ▶ Usually represented by a '?' symbol:

$$c_1 c_2 c_3 \dots c_n \longrightarrow \boxed{\text{Channel}} \longrightarrow c_1 ? c_3 ?? c_6 \dots c_n$$

Proposition:

Let \mathcal{C} be an $[n, k, d]$ code over \mathbb{F} . There is a decoder for \mathcal{C} that corrects any occurrence of up to $d - 1$ erasures.

Proof: Let $\Phi = \mathbb{F} \cup \{?\}$.

Consider the decoder defined for each $\mathbf{y} \in \Phi^n$ as

$$\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{c} & \text{if } \mathbf{c} \text{ is the } \underline{\text{unique}} \text{ codeword that agrees with } \mathbf{y} \\ & \text{on all unerased positions} \\ \text{ERROR} & \text{otherwise.} \end{cases}$$

Erasures

Proof (cont'd):

- ▶ Suppose that \mathbf{c} was transmitted and at most $d - 1$ of its coordinates were erased.
- ▶ The received word \mathbf{y} contains at most $d - 1$ '?' symbols, and agrees with \mathbf{c} on all the unerased positions.
- ▶ If there were another $\mathbf{c}' \in \mathcal{C}$ that also agreed with \mathbf{y} on all the unerased positions, then \mathbf{c} and \mathbf{c}' could differ only in those positions where \mathbf{y} has '?' symbols.

\mathbf{y}	_____	?	?	...	?	_____
\mathbf{c}	_____	*	*	...	*	_____
\mathbf{c}'	_____	□	□	...	□	_____

Then, $d(\mathbf{c}, \mathbf{c}') \leq d - 1$, which contradicts $d_{\min}(\mathcal{C}) = d$.

- ▶ Thus, \mathbf{c} is the unique codeword that agrees with \mathbf{y} in all unerased coordinates, and hence, $\mathcal{D}(\mathbf{y}) = \mathbf{c}$.

Generator and Parity-Check Matrices

An $[n, k]$ linear code \mathcal{C} is specified by one of the following —

- ▶ **Generator matrix:** This is a $k \times n$ matrix G whose rows contain a **basis** for \mathcal{C} .
- ▶ **Parity-check matrix:** This is an $m \times n$ matrix H such that

$$\mathcal{C} = \ker(H) = \{\mathbf{x} \in \mathbb{F}^n : H\mathbf{x}^T = \mathbf{0}\}$$

Notes:

- ▶ $\dim(\mathcal{C}) = n - \text{rank}(H)$
- ▶ $HG^T = \mathbf{0}$
- ▶ If $H = [A \mid I_{n-k}]$ is a p.c. matrix, then $G = [I_k \mid -A^T]$ is a gen. matrix.

Example: The $[7, 4]$ Hamming Code

The $[7, 4]$ binary Hamming code has parity-check matrix

$$H = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I_3},$$

Hence,

$$G = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{I_4} \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{-A^T},$$

is a generator matrix for the code. (Note that $-A^T = A^T$ over \mathbb{F}_2 .)

Generator Matrices and Encoding

An $[n, k]$ linear code \mathcal{C} over \mathbb{F}_q has q^k codewords.

So, there is a 1-1 correspondence between \mathbb{F}_q^k and $\mathcal{C} \subseteq \mathbb{F}_q^n$.

An **encoder** for \mathcal{C} is a 1-1 mapping from **message words**

$\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{F}_q^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator Matrices and Encoding

An $[n, k]$ linear code \mathcal{C} over \mathbb{F}_q has q^k codewords.

So, there is a 1-1 correspondence between \mathbb{F}_q^k and $\mathcal{C} \subseteq \mathbb{F}_q^n$.

An **encoder** for \mathcal{C} is a 1-1 mapping from **message words**

$\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{F}_q^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator matrices give rise to encoders:

- ▶ Let G be a $k \times n$ generator matrix for \mathcal{C} . Its rows $\mathbf{g}_1, \dots, \mathbf{g}_k$ form a basis of \mathcal{C} .
- ▶ Recall that every $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination $\sum_{j=1}^k u_j \mathbf{g}_j$, with $u_j \in \mathbb{F}_q$ for all j .

Generator Matrices and Encoding

An $[n, k]$ linear code \mathcal{C} over \mathbb{F}_q has q^k codewords.

So, there is a 1-1 correspondence between \mathbb{F}_q^k and $\mathcal{C} \subseteq \mathbb{F}_q^n$.

An **encoder** for \mathcal{C} is a 1-1 mapping from **message words**

$\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{F}_q^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator matrices give rise to encoders:

- ▶ Let G be a $k \times n$ generator matrix for \mathcal{C} . Its rows $\mathbf{g}_1, \dots, \mathbf{g}_k$ form a basis of \mathcal{C} .
- ▶ Recall that every $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination $\sum_{j=1}^k u_j \mathbf{g}_j$, with $u_j \in \mathbb{F}_q$ for all j .
- ▶ Hence, the mapping

$$\mathbf{u} = \underbrace{(u_1, \dots, u_k)}_{\in \mathbb{F}_q^k} \mapsto \mathbf{u} G = \sum_{j=1}^k u_j \mathbf{g}_j$$

is a bijection between \mathbb{F}_q^k and \mathcal{C} , i.e., an encoder mapping.

Systematic Generator Matrices

G is called a **systematic** generator matrix if it is of the form

$$G = [I_k \mid B],$$

where I_k is the $k \times k$ identity matrix and B is a $k \times (n - k)$ matrix.

In such a case, the encoder $\mathbf{u} \mapsto \mathbf{u}G$ maps a message $\mathbf{u} \in \mathbb{F}_q^k$ to the codeword

$$\mathbf{c} = \left[\underbrace{\mathbf{u}}_{k \text{ symbols}} \mid \underbrace{\mathbf{u}B}_{n - k \text{ symbols}} \right].$$

- ▶ In \mathbf{c} , the first k symbols constitute the message \mathbf{u} itself; they are called **information symbols**.
- ▶ The remaining $n - k$ symbols, $\mathbf{u}B$, are **parity-check symbols**.

So, retrieving the message encoded within a codeword is easy.

Example: The $[7, 4]$ Hamming Code

A (systematic) generator matrix for the $[7, 4]$ Hamming code is

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

Using this generator matrix, $\mathbf{u} = [u_1, u_2, u_3, u_4] \in (\mathbb{F}_2)^4$ gets mapped to the codeword

$$[u_1, u_2, u_3, u_4, u_1 \oplus u_2 \oplus u_4, u_1 \oplus u_3 \oplus u_4, u_2 \oplus u_3 \oplus u_4]$$

Dual Codes

\mathcal{C} an $[n, k]$ linear code over a field \mathbb{F} .

For $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathbb{F}^n$, let

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + \dots + x_n y_n,$$

all operations taking place within the field \mathbb{F} .

Definition

The **dual code** of \mathcal{C} is defined to be

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}^n : \mathbf{x} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C}\}$$

- ▶ \mathcal{C}^\perp is an $[n, n - k]$ code
- ▶ Any generator matrix for \mathcal{C} is a parity-check matrix for \mathcal{C}^\perp .
- ▶ Any generator matrix for \mathcal{C}^\perp is a parity-check matrix for \mathcal{C} .

Example: The $[7, 4]$ Hamming Code

Let \mathcal{C} be the $[7, 4]$ Hamming code.

The dual code, \mathcal{C}^\perp , is the $[7, 3]$ binary linear code **generated** by

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

Example: The $[7, 4]$ Hamming Code

Let \mathcal{C} be the $[7, 4]$ Hamming code.

The dual code, \mathcal{C}^\perp , is the $[7, 3]$ binary linear code generated by

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

- ▶ Observe that $HH^T = 0$ (over \mathbb{F}_2).
- ▶ Thus, every row of H is in the nullspace of H , or equivalently,

$$\underbrace{\text{rowspace}(H)}_{\mathcal{C}^\perp} \subset \underbrace{\text{nullspace}(H)}_{\mathcal{C}}$$

- ▶ Such a code \mathcal{C} is said to be **dual-containing**.

Encoding

Given: a linear code \mathcal{C} with generator matrix G .

The mapping $\mathcal{E} : \mathbb{F}^k \rightarrow \mathcal{C}$ defined by

$$\mathbf{u} \mapsto \mathbf{u}G$$

is an **encoder mapping** for \mathcal{C} .

Decoding

Given:

- ▶ Linear code \mathcal{C} of length n
- ▶ Probabilistic channel:

$$\Pr[\mathbf{y} \mid \mathbf{x}] := \text{Prob}[\mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted}]$$

- ▶ Received word: $\mathbf{y} = (y_1, y_2, \dots, y_n)$

Maximum-Likelihood (ML) Decoding:

$$\text{Decode to } \hat{\mathbf{c}} = \arg \max_{\mathbf{x} \in \mathcal{C}} \Pr[\mathbf{y} \mid \mathbf{x}]$$

Decoding

Given:

- ▶ Linear code \mathcal{C} of length n
- ▶ Probabilistic channel:

$$\Pr[\mathbf{y} \mid \mathbf{x}] := \text{Prob}[\mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted}]$$

- ▶ Received word: $\mathbf{y} = (y_1, y_2, \dots, y_n)$

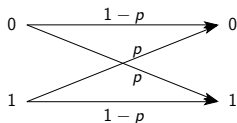
Maximum-Likelihood (ML) Decoding:

$$\text{Decode to } \hat{\mathbf{c}} = \arg \max_{\mathbf{x} \in \mathcal{C}} \Pr[\mathbf{y} \mid \mathbf{x}]$$

Assuming all codewords are *a priori* equally likely,
ML decoding minimizes probability of decoding error.

Minimum Distance Decoding

Over certain channels, e.g.,
the memoryless **binary symmetric channel**, ML decoding reduces to

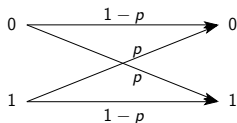


Minimum Distance (MD) Decoding:

Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{y}, \mathbf{c})$.

Minimum Distance Decoding

Over certain channels, e.g.,
the memoryless **binary symmetric channel**, ML decoding reduces to



Minimum Distance (MD) Decoding:

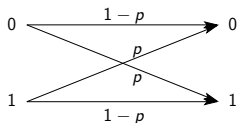
Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{y}, \mathbf{c})$.

We exploit linearity to give some different perspectives on MDD.

Note that $d(\mathbf{y}, \mathbf{c}) = w(\mathbf{y} - \mathbf{c})$.

Minimum Distance Decoding

Over certain channels, e.g.,
the memoryless **binary symmetric channel**, ML decoding reduces to



Minimum Distance (MD) Decoding:

Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{y}, \mathbf{c})$.

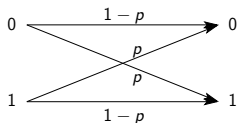
We exploit linearity to give some different perspectives on MDD.

Note that $d(\mathbf{y}, \mathbf{c}) = w(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =: \text{error vector } \mathbf{e}$

Minimum Distance Decoding

Over certain channels, e.g.,
the memoryless **binary symmetric channel**, ML decoding reduces to



Minimum Distance (MD) Decoding:

Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{y}, \mathbf{c})$.

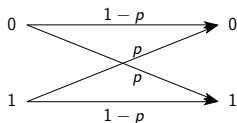
We exploit linearity to give some different perspectives on MDD.

Note that $d(\mathbf{y}, \mathbf{c}) = w(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =:$ **error vector** $\mathbf{e} \iff \boxed{\mathbf{c} = \mathbf{y} - \mathbf{e}}$

Minimum Distance Decoding

Over certain channels, e.g.,
the memoryless **binary symmetric channel**, ML decoding reduces to



Minimum Distance (MD) Decoding:

Decode \mathbf{y} to closest codeword: $\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{y}, \mathbf{c})$.

We exploit linearity to give some different perspectives on MDD.

Note that $d(\mathbf{y}, \mathbf{c}) = w(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =:$ **error vector** $\mathbf{e} \iff \boxed{\mathbf{c} = \mathbf{y} - \mathbf{e}}$

So, the MDD rule is equivalent to the following:

Given a received vector $\mathbf{y} \in \mathbb{F}^n$, find an error vector \mathbf{e} of least Hamming weight such that $\mathbf{y} - \mathbf{e} \in \mathcal{C}$.

Decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$.

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to \mathbf{y} .)

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to \mathbf{y} .)

Note that

$$\begin{aligned}\mathbf{e} \in \mathcal{E}(\mathbf{y}) &\iff \mathbf{y} - \mathbf{e} = \mathbf{c}' \quad \text{for some } \mathbf{c}' \in \mathcal{C} \\ &\iff \mathbf{y} - \mathbf{e} = -\mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \\ &\iff \mathbf{e} = \mathbf{y} + \mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C}\end{aligned}$$

Hence,

$$\mathcal{E}(\mathbf{y}) = \{\mathbf{y} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\} =: \mathbf{y} + \mathcal{C}.$$

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to \mathbf{y} .)

Note that

$$\begin{aligned}\mathbf{e} \in \mathcal{E}(\mathbf{y}) &\iff \mathbf{y} - \mathbf{e} = \mathbf{c}' \quad \text{for some } \mathbf{c}' \in \mathcal{C} \\ &\iff \mathbf{y} - \mathbf{e} = -\mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \\ &\iff \mathbf{e} = \mathbf{y} + \mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C}\end{aligned}$$

Hence,

$$\mathcal{E}(\mathbf{y}) = \{\mathbf{y} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\} =: \mathbf{y} + \mathcal{C}.$$

Thus, $\mathcal{E}(\mathbf{y})$ is a **coset** of \mathcal{C} .

Another Perspective on MDD

MDD can now be viewed as the following algorithm: Given a received vector $\mathbf{y} \in \mathbb{F}^n$,

1. find the coset of \mathcal{C} to which \mathbf{y} belongs
2. identify a vector, \mathbf{e} , of least weight from that coset
3. set $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Another Perspective on MDD

MDD can now be viewed as the following algorithm: Given a received vector $\mathbf{y} \in \mathbb{F}^n$,

1. find the coset of \mathcal{C} to which \mathbf{y} belongs
2. identify a vector, \mathbf{e} , of least weight from that coset
3. set $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Some advantages offered by this perspective on MDD:

- ▶ All cosets of \mathcal{C} can be pre-calculated and stored at the decoder. This pre-computation has to be done just once, and does not have to be repeated each time a new \mathbf{y} is received.
- ▶ A vector of least weight within each coset, called a **coset leader**, can also be identified in advance and stored.

Cosets — A Quick Review

Definition: A **coset** of \mathcal{C} in \mathbb{F}^n is a set of the form
 $\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\}$, for some $\mathbf{b} \in \mathbb{F}^n$.

Cosets — A Quick Review

Definition: A **coset** of \mathcal{C} in \mathbb{F}^n is a set of the form
$$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\}, \text{ for some } \mathbf{b} \in \mathbb{F}^n.$$

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between \mathcal{C} and $\mathbf{b} + \mathcal{C}$. □

Cosets — A Quick Review

Definition: A **coset** of \mathcal{C} in \mathbb{F}^n is a set of the form
$$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\}, \text{ for some } \mathbf{b} \in \mathbb{F}^n.$$

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between \mathcal{C} and $\mathbf{b} + \mathcal{C}$. □

2. Each $\mathbf{b} \in \mathbb{F}^n$ lies in some coset of \mathcal{C} .

Proof: Clearly, $\mathbf{b} \in \mathbf{b} + \mathcal{C}$, since $\mathbf{0} \in \mathcal{C}$. □

Cosets — A Quick Review

Definition: A **coset** of \mathcal{C} in \mathbb{F}^n is a set of the form
$$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\}, \text{ for some } \mathbf{b} \in \mathbb{F}^n.$$

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between \mathcal{C} and $\mathbf{b} + \mathcal{C}$. □

2. Each $\mathbf{b} \in \mathbb{F}^n$ lies in some coset of \mathcal{C} .

Proof: Clearly, $\mathbf{b} \in \mathbf{b} + \mathcal{C}$, since $\mathbf{0} \in \mathcal{C}$. □

3. \mathbf{a} and \mathbf{b} are in the same coset of \mathcal{C} iff $\mathbf{a} - \mathbf{b} \in \mathcal{C}$.

Proof: Suppose that $\mathbf{b} \in \mathbf{y} + \mathcal{C}$, so that $\mathbf{b} = \mathbf{y} + \mathbf{c}$ for some $\mathbf{c} \in \mathcal{C}$.

Then, $\mathbf{a} = \mathbf{b} + (\mathbf{a} - \mathbf{b}) = \mathbf{y} + (\mathbf{c} + \mathbf{a} - \mathbf{b})$.

Hence, \mathbf{a} is also in $\mathbf{y} + \mathcal{C} \iff \mathbf{c} + (\mathbf{a} - \mathbf{b}) \in \mathcal{C}$

$\iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$. □

Cosets — A Quick Review

Theorem: The distinct cosets of a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ form a **partition** of \mathbb{F}^n .

Proof: Define a relation \sim on \mathbb{F}^n as follows: for $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$,

$$\mathbf{a} \sim \mathbf{b} \iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$$

- ▶ It is easy to verify that \sim is an equivalence relation.
- ▶ Hence, the equivalence classes of \sim form a partition of \mathbb{F}^n .
- ▶ However, by Basic Fact 3, the equivalence classes of \sim are precisely the cosets of \mathcal{C} . □

Cosets — A Quick Review

Theorem: The distinct cosets of a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ form a **partition** of \mathbb{F}^n .

Proof: Define a relation \sim on \mathbb{F}^n as follows: for $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$,

$$\mathbf{a} \sim \mathbf{b} \iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$$

- ▶ It is easy to verify that \sim is an equivalence relation.
- ▶ Hence, the equivalence classes of \sim form a partition of \mathbb{F}^n .
- ▶ However, by Basic Fact 3, the equivalence classes of \sim are precisely the cosets of \mathcal{C} . □

Corollary: An $[n, k]$ linear code over \mathbb{F}_q has q^{n-k} cosets.

Proof: Each coset has q^k words (by Basic Fact 1).

Put together, the cosets partition \mathbb{F}_q^n .

Hence, the number of cosets is $q^n / q^k = q^{n-k}$. □

Back to MDD

\mathcal{C} an $[n, k]$ linear code over \mathbb{F}_q .

Pre-computation (one-time):

- ▶ List out all q^{n-k} cosets of \mathcal{C} .
- ▶ Identify a coset leader (word of least weight) from each coset.

Given a received vector $\mathbf{y} \in \mathbb{F}_q^n$,

1. find the coset of \mathcal{C} to which \mathbf{y} belongs
2. retrieve the coset leader, \mathbf{e} , identified for that coset
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Back to MDD

\mathcal{C} an $[n, k]$ linear code over \mathbb{F}_q .

Pre-computation (one-time):

- ▶ List out all q^{n-k} cosets of \mathcal{C} .
- ▶ Identify a coset leader (word of least weight) from each coset.

Given a received vector $\mathbf{y} \in \mathbb{F}_q^n$,

1. find the coset of \mathcal{C} to which \mathbf{y} belongs
2. retrieve the coset leader, \mathbf{e} , identified for that coset
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$


Example: Let \mathcal{C} be the $[6, 3]$ binary linear code generated by

$$G = \left[\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I_3} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \right]$$

\mathcal{C} has $2^{n-k} = 8$ cosets. It can be verified that $d_{\min}(\mathcal{C}) = 3$.


The Standard Array

A **standard array** for a linear code is a listing of the code and all its cosets in the form of an array.

\mathcal{C}	=	000000	100110	010101	001011	111000	011110	101101	110011
$100000 + \mathcal{C}$	=	100000	000110	110101	101011	011000	111110	001101	010011
$010000 + \mathcal{C}$	=	010000	110110	000101	011011	101000	001110	111101	100011
$001000 + \mathcal{C}$	=	001000	101110	011101	000011	110000	010110	100101	111011
$000100 + \mathcal{C}$	=	000100	100010	010001	001111	111100	011010	101001	110111
$000010 + \mathcal{C}$	=	000010	100100	010111	001001	111010	011100	101111	110001
$000001 + \mathcal{C}$	=	000001	100111	010100	001010	111001	011111	101100	110010
$100001 + \mathcal{C}$	=	100001	000111	110100	101010	011001	111111	001100	010010
									
		coset leaders							

The Standard Array

A **standard array** for a linear code is a listing of the code and all its cosets in the form of an array.

\mathcal{C}	=	000000	100110	010101	001011	111000	011110	101101	110011
$100000 + \mathcal{C}$	=	100000	000110	110101	101011	011000	111110	001101	010011
$010000 + \mathcal{C}$	=	010000	110110	000101	011011	101000	001110	111101	100011
$001000 + \mathcal{C}$	=	001000	101110	011101	000011	110000	010110	100101	111011
$000100 + \mathcal{C}$	=	000100	100010	010001	001111	111100	011010	101001	110111
$000010 + \mathcal{C}$	=	000010	100100	010111	001001	111010	011100	101111	110001
$000001 + \mathcal{C}$	=	000001	100111	010100	001010	111001	011111	101100	110010
$100001 + \mathcal{C}$	=	100001	000111	110100	101010	011001	111111	001100	010010
		 coset leaders							

Note: There may be multiple choices for coset leader: for example, in the last coset, we could have alternatively chosen 001100 or 010010 as coset leaders.

Unique Coset Leader

When is there a unique coset leader?

Note that any word of Hamming weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ is always the unique word of least weight within its coset.

[If two words **a**, **b** of weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ were in the same coset, then their difference **a** – **b** would be a word of weight $< d_{\min}$ in \mathcal{C} .]

Unique Coset Leader

When is there a unique coset leader?

Note that any word of Hamming weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ is always the unique word of least weight within its coset.

[If two words **a**, **b** of weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ were in the same coset, then their difference **a** – **b** would be a word of weight $< d_{\min}$ in \mathcal{C} .]

Example: In the previous example of the $[6, 3, 3]$ linear code,

- ▶ All words of weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor = 1$ are the unique coset leaders of their respective cosets.

Correctable Error Patterns

The coset leaders are precisely the error patterns that get corrected by the standard array implementation of MDD.

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- ▶ Suppose $\mathbf{y} = 110110$ is received. So, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 010000$ is the error vector (but this is not *a priori* known to the decoder).

Correctable Error Patterns

The coset leaders are precisely the error patterns that get corrected by the standard array implementation of MDD.

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- ▶ Suppose $\mathbf{y} = 110110$ is received. So, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 010000$ is the error vector (but this is not *a priori* known to the decoder).
- ▶ \mathbf{y} is in the coset $010000 + \mathcal{C}$, for which 010000 , being the unique word of least weight, is the coset leader.
- ▶ So, \mathbf{y} gets decoded to $\hat{\mathbf{c}} = \mathbf{y} - \text{coset leader} = 100110 = \mathbf{c}$.
The error vector gets corrected!
Note that this \mathbf{c} is the unique closest codeword to \mathbf{y} .

Correctable Error Patterns

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- ▶ Suppose $\mathbf{y} = 101010$ is received. Now, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 001100$ is the error vector (again, not *a priori* known to the decoder).

Correctable Error Patterns

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- ▶ Suppose $\mathbf{y} = 101010$ is received. Now, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 001100$ is the error vector (again, not *a priori* known to the decoder).
- ▶ \mathbf{y} is in the coset $100001 + \mathcal{C}$, for which 100001 was chosen to be the coset leader in our standard array.
- ▶ So, \mathbf{y} gets decoded to $\hat{\mathbf{c}} = \mathbf{y} - \text{coset leader} = 001011 \neq \mathbf{c}$. The error vector does not get corrected this time.

Note that this $\hat{\mathbf{c}}$ is a closest codeword to \mathbf{y} , but it is not the unique such codeword:

- ▶ There are three words of weight 2 in the same coset as \mathbf{y} , including the actual error vector $\mathbf{e} = 100001$.
- ▶ Subtracting any of these weight-2 words from \mathbf{y} would yield a codeword at distance 2 from \mathbf{y} .

Storage Complexity

- ▶ Even the one-time pre-computation and storage of the standard array, which contains all the q^n words in \mathbb{F}_q^n , is infeasible for $n \sim 100$ or more.
- ▶ Storage of the entire array is not needed if we can find some means of identifying the coset to which a given $\mathbf{y} \in \mathbb{F}_q^n$ belongs.

It would then suffice to store only the q^{n-k} coset leaders.

Syndromes

Let H be an $(n - k) \times n$ parity-check matrix for \mathcal{C} .

Definition: The **syndrome** of a vector $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

- ▶ \mathbf{s} is an $(n - k) \times 1$ column vector
- ▶ The syndrome of $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.

Syndromes

Let H be an $(n - k) \times n$ parity-check matrix for \mathcal{C} .

Definition: The **syndrome** of a vector $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

- ▶ \mathbf{s} is an $(n - k) \times 1$ column vector
- ▶ The syndrome of $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.
- ▶ Two vectors $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}^n$ are in the same coset of \mathcal{C} iff they have the same syndrome.

Proof:

$$\begin{aligned}\mathbf{y}_1, \mathbf{y}_2 \text{ are in the same coset of } \mathcal{C} &\iff \mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C} \\ &\iff H(\mathbf{y}_1 - \mathbf{y}_2)^T = \mathbf{0} \\ &\iff H\mathbf{y}_1^T = H\mathbf{y}_2^T \quad \square\end{aligned}$$

Syndromes

Let H be an $(n - k) \times n$ parity-check matrix for \mathcal{C} .

Definition: The **syndrome** of a vector $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

- ▶ \mathbf{s} is an $(n - k) \times 1$ column vector
- ▶ The syndrome of $\mathbf{y} \in \mathbb{F}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.
- ▶ Two vectors $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}^n$ are in the same coset of \mathcal{C} iff they have the same syndrome.

Proof:

$$\begin{aligned}\mathbf{y}_1, \mathbf{y}_2 \text{ are in the same coset of } \mathcal{C} &\iff \mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C} \\ &\iff H(\mathbf{y}_1 - \mathbf{y}_2)^T = \mathbf{0} \\ &\iff H\mathbf{y}_1^T = H\mathbf{y}_2^T \quad \square\end{aligned}$$

Thus, the syndrome of a word $\mathbf{y} \in \mathbb{F}^n$ uniquely determines the coset to which it belongs.

Syndrome Decoding

Therefore, to implement MDD, it is enough to store a list of coset leaders along with the syndromes for their respective cosets.

In all, this requires storing q^{n-k} (coset leader, syndrome) pairs.

MDD then reduces to **syndrome decoding**: Given a rcvd $\mathbf{y} \in \mathbb{F}_q^n$,

1. compute $\mathbf{s} = H\mathbf{y}^T$
2. retrieve the coset leader, \mathbf{e} , corresponding to syndrome \mathbf{s}
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Example

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

We then form a table of syndromes and corresp. coset leaders:

<u>Coset leader, \mathbf{e}</u>	<u>Syndrome $\mathbf{s} = H\mathbf{e}^T$</u>
000000	$[0 \ 0 \ 0]^T$
100000	$[1 \ 1 \ 0]^T$
010000	$[1 \ 0 \ 1]^T$
001000	$[0 \ 1 \ 1]^T$
000100	$[1 \ 0 \ 0]^T$
000010	$[0 \ 1 \ 0]^T$
000001	$[0 \ 0 \ 1]^T$
100001	$[1 \ 1 \ 1]^T$

Example

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

We then form a table of syndromes and corresp. coset leaders:

<u>Coset leader, \mathbf{e}</u>	<u>Syndrome $\mathbf{s} = H\mathbf{e}^T$</u>
000000	$[0 \ 0 \ 0]^T$
100000	$[1 \ 1 \ 0]^T$
010000	$[1 \ 0 \ 1]^T$
001000	$[0 \ 1 \ 1]^T$
000100	$[1 \ 0 \ 0]^T$
000010	$[0 \ 1 \ 0]^T$
000001	$[0 \ 0 \ 1]^T$
100001	$[1 \ 1 \ 1]^T$

To illustrate syndrome decoding,

$$\begin{aligned} \mathbf{y} = 110110 &\implies \mathbf{s} = H\mathbf{y}^T = [1 \ 0 \ 1]^T \\ &\implies \mathbf{e} = 010000 \implies \hat{\mathbf{c}} = \mathbf{y} - \mathbf{e} = 100110 \end{aligned}$$

The Challenge

It is known that MD decoding (for an arbitrary linear code) is **NP-hard** [Berlekamp, McEliece, van Tilborg (1978)]

The Challenge

It is known that MD decoding (for an arbitrary linear code) is **NP-hard** [Berlekamp, McEliece, van Tilborg (1978)]

The main challenge of coding theory is to design linear code families that

- ▶ have good error-correcting properties
- ▶ are efficiently decodable

t -Error-Correcting Codes

Recall: A code \mathcal{C} is **t -error-correcting** if there exists a decoding map $\mathcal{D} : \mathbb{F}^n \rightarrow \mathcal{C}$ such that
whenever $d(\mathbf{y}, \mathbf{c}) \leq t$, we have $\mathcal{D}(\mathbf{y}) = \mathbf{c}$.

t -Error-Correcting Codes

Recall: A code \mathcal{C} is **t -error-correcting** if there exists a decoding map $\mathcal{D} : \mathbb{F}^n \rightarrow \mathcal{C}$ such that
whenever $d(\mathbf{y}, \mathbf{c}) \leq t$, we have $\mathcal{D}(\mathbf{y}) = \mathbf{c}$.

Equivalent definitions:

- ▶ A code is t -error-correcting if all error patterns \mathbf{e} of weight $w_H(\mathbf{e}) \leq t$ can be corrected under minimum distance decoding
- ▶ A code is t -error-correcting iff all the distinct error vectors of weight $\leq t$ can be chosen to be coset leaders of distinct cosets (or equivalently, all these error vectors lie in distinct cosets).

t -Error-Correcting Codes

Recall: A code \mathcal{C} is t -error-correcting if there exists a decoding map $\mathcal{D} : \mathbb{F}^n \rightarrow \mathcal{C}$ such that
whenever $d(\mathbf{y}, \mathbf{c}) \leq t$, we have $\mathcal{D}(\mathbf{y}) = \mathbf{c}$.

Equivalent definitions:

- ▶ A code is t -error-correcting if all error patterns \mathbf{e} of weight $w_H(\mathbf{e}) \leq t$ can be corrected under minimum distance decoding
- ▶ A code is t -error-correcting iff all the distinct error vectors of weight $\leq t$ can be chosen to be coset leaders of distinct cosets (or equivalently, all these error vectors lie in distinct cosets).

Thus,

- ▶ A linear code with minimum distance d is t -error-correcting for any $t \leq \frac{d-1}{2}$.

The Hamming Bound

Theorem:

If an $[n, k]$ linear code over \mathbb{F}_q is t -error-correcting, then

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

The Hamming Bound

Theorem:

If an $[n, k]$ linear code over \mathbb{F}_q is t -error-correcting, then

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

Proof:

- ▶ LHS = no. of error vectors of weight $\leq t$
- ▶ RHS = no. of cosets



The Hamming Bound

Theorem:

If an $[n, k]$ linear code over \mathbb{F}_q is t -error-correcting, then

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

Proof:

- ▶ LHS = no. of error vectors of weight $\leq t$
- ▶ RHS = no. of cosets



Definition: A t -error-correcting linear code whose parameters satisfy the Hamming bound with equality is called a **perfect code**.

- ▶ Such a code, under MDD, can correct **all** error patterns of weight $\leq t$, but **none** of weight $t+1$ or more.

Examples of Perfect Codes

- ▶ The $[n, 1, n]$ binary repetition code. It is a perfect $\frac{n-1}{2}$ -error-correcting code, for odd values of n :

$$\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} = \frac{1}{2} \sum_{i=0}^n \binom{n}{i} = 2^{n-1} = 2^{n-k}.$$

Examples of Perfect Codes

- ▶ The $[n, 1, n]$ binary repetition code. It is a perfect $\frac{n-1}{2}$ -error-correcting code, for odd values of n :

$$\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} = \frac{1}{2} \sum_{i=0}^n \binom{n}{i} = 2^{n-1} = 2^{n-k}.$$

- ▶ The $[7, 4, 3]$ binary Hamming code is a perfect 1-error-correcting code:

$$\binom{7}{0} + \binom{7}{1} = 8 = 2^{7-4}.$$

This construction can be generalized to obtain a family of perfect single-error-correcting codes: the $[2^r - 1, 2^r - r - 1, 3]$ binary Hamming codes.

The Singleton Bound

Theorem: For any $[n, k, d]$ linear code over \mathbb{F}_q , we have

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

Proof: Let H be a parity-check matrix of an $[n, k]$ linear code.
Then,

$$\text{rank}(H) = n - k$$

\implies any set of $n - k + 1$ columns of H is linearly dependent

\implies there exists some codeword of weight $\leq n - k + 1$

$\implies d_{\min} \leq n - k + 1.$



The Singleton Bound

Theorem: For any $[n, k, d]$ linear code over \mathbb{F}_q , we have

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

Proof: Let H be a parity-check matrix of an $[n, k]$ linear code.
Then,

$$\text{rank}(H) = n - k$$

\implies any set of $n - k + 1$ columns of H is linearly dependent

\implies there exists some codeword of weight $\leq n - k + 1$

$\implies d_{\min} \leq n - k + 1.$



Definition: An $[n, k, d]$ linear code that satisfies $d = n - k + 1$ is called a **maximum distance separable (MDS)** code.

Examples of MDS Codes

The following are all examples of MDS codes, over any field \mathbb{F} :

- ▶ \mathbb{F}^n , which is an $[n, n, 1]$ code
- ▶ the single parity-check code, defined by the parity-check matrix $H = [1 \ 1 \ \dots \ 1]$ — this is an $[n, n - 1, 2]$ code
- ▶ the $[n, 1, n]$ repetition code, generated by $G = [1 \ 1 \ \dots \ 1]$

Examples of MDS Codes

The following are all examples of MDS codes, over any field \mathbb{F} :

- ▶ \mathbb{F}^n , which is an $[n, n, 1]$ code
- ▶ the single parity-check code, defined by the parity-check matrix $H = [1 \ 1 \ \dots \ 1]$ — this is an $[n, n - 1, 2]$ code
- ▶ the $[n, 1, n]$ repetition code, generated by $G = [1 \ 1 \ \dots \ 1]$

The above are, in fact, the only MDS codes possible when $\mathbb{F} = \mathbb{F}_2$.

Over non-binary fields, we have a wide variety of other examples; a particularly important one is the family of **Reed-Solomon codes**.

Examples of MDS Codes

The following are all examples of MDS codes, over any field \mathbb{F} :

- ▶ \mathbb{F}^n , which is an $[n, n, 1]$ code
- ▶ the single parity-check code, defined by the parity-check matrix $H = [1 \ 1 \ \dots \ 1]$ — this is an $[n, n - 1, 2]$ code
- ▶ the $[n, 1, n]$ repetition code, generated by $G = [1 \ 1 \ \dots \ 1]$

The above are, in fact, the only MDS codes possible when $\mathbb{F} = \mathbb{F}_2$.

Over non-binary fields, we have a wide variety of other examples; a particularly important one is the family of **Reed-Solomon codes**.

MDS codes have some interesting properties. For example,

- ▶ \mathcal{C} is MDS iff \mathcal{C}^\perp is MDS.

The Gilbert-Varshamov (GV) Bound

The GV bound is a **sufficient condition** on code parameters that guarantees the existence of a linear code with those parameters.

The Gilbert-Varshamov (GV) Bound

The GV bound is a **sufficient condition** on code parameters that guarantees the existence of a linear code with those parameters.

Theorem (The Gilbert-Varshamov bound)

Let \mathbb{F}_q be a finite field, and let n , k and d be positive integers such that

$$\sum_{\ell=0}^{d-2} \binom{n-1}{\ell} (q-1)^\ell < q^{n-k}.$$

Then, there exists an $[n, k]$ linear code over \mathbb{F}_q , with $d_{\min} \geq d$.

Proof of GV Bound

The idea is to construct an $(n - k) \times n$ parity-check matrix H with the property that no $d - 1$ (or fewer) columns are linearly dependent over \mathbb{F}_q .

The construction is recursive:

- ▶ Pick the first column \mathbf{h}_1 to be any non-zero vector in \mathbb{F}_q^{n-k} .
- ▶ Suppose that, for some $i \geq 2$, we have picked the first $i - 1$ columns $\mathbf{h}_1, \dots, \mathbf{h}_{i-1}$.

We then pick \mathbf{h}_i so that it cannot be obtained as a linear combination of any $d - 2$ (or fewer) columns from $\mathbf{h}_1, \dots, \mathbf{h}_{i-1}$.

Proof of GV Bound

The idea is to construct an $(n - k) \times n$ parity-check matrix H with the property that no $d - 1$ (or fewer) columns are linearly dependent over \mathbb{F}_q .

The construction is recursive:

- ▶ Pick the first column \mathbf{h}_1 to be any non-zero vector in \mathbb{F}_q^{n-k} .
- ▶ Suppose that, for some $i \geq 2$, we have picked the first $i - 1$ columns $\mathbf{h}_1, \dots, \mathbf{h}_{i-1}$.

We then pick \mathbf{h}_i so that it cannot be obtained as a linear combination of any $d - 2$ (or fewer) columns from $\mathbf{h}_1, \dots, \mathbf{h}_{i-1}$.

It can be shown that if the inequality in the GV bound is satisfied, then an $(n - k) \times n$ matrix $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_n]$ can be constructed.

Binary Hamming Codes

Let $r \geq 1$ be an integer.

The **binary Hamming code** \mathcal{H}_r is the binary linear code specified by an $r \times n$ parity-check matrix whose columns are all the distinct, non-zero binary r -tuples: $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_{2^r-1}]$

► Example ($r = 3$):

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

Binary Hamming Codes

Let $r \geq 1$ be an integer.

The **binary Hamming code** \mathcal{H}_r is the binary linear code specified by an $r \times n$ parity-check matrix whose columns are all the distinct, non-zero binary r -tuples: $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_{2^r-1}]$

- ▶ Example ($r = 3$):

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

- ▶ Blocklength, $n = 2^r - 1$
- ▶ $\text{rank}(H) = r \implies \dim(\mathcal{H}_r) = n - r = 2^r - 1 - r$
- ▶ all columns are distinct and non-zero, but some three columns sum to 0 $\implies d_{\min}(\mathcal{H}_r) = 3$.
- ▶ Thus, \mathcal{H}_r is an $[2^r - 1, 2^r - 1 - r, 3]$ binary linear code.

Binary Reed-Muller Codes

- ▶ Set $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.
- ▶ Recursively define the matrices (tensor powers) $G^{\otimes n}$, $n = 1, 2, \dots$, as:

- ▶ $G^{\otimes 1} = G$

- ▶ $G^{\otimes m} = \begin{bmatrix} G^{\otimes(m-1)} & 0 \\ G^{\otimes(m-1)} & G^{\otimes(m-1)} \end{bmatrix}$

$G^{\otimes m}$ is a $2^m \times 2^m$ matrix, called a **Hadamard matrix**.

Binary Reed-Muller Codes

- ▶ Set $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.
- ▶ Recursively define the matrices (tensor powers) $G^{\otimes n}$, $n = 1, 2, \dots$, as:
 - ▶ $G^{\otimes 1} = G$
 - ▶ $G^{\otimes m} = \begin{bmatrix} G^{\otimes(m-1)} & 0 \\ G^{\otimes(m-1)} & G^{\otimes(m-1)} \end{bmatrix}$

$G^{\otimes m}$ is a $2^m \times 2^m$ matrix, called a **Hadamard matrix**.

- ▶ The **r -th order Reed-Muller code $RM(m, r)$** is the binary linear code generated by the submatrix of $G^{\otimes m}$ formed by the rows of Hamming weight $\geq 2^{m-r}$.

An Example: $RM(3, 2)$

$RM(3, 2)$ is generated by the rows with Hamming weight at least $2^{3-2} = 2$, of the matrix below:

$$G^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Some Properties of $\text{RM}(m, r)$

- ▶ Code Parameters:
 - ▶ Blocklength, $n = 2^m$
 - ▶ Dimension, $k = \sum_{i=0}^r \binom{m}{i}$
 - ▶ Minimum distance, $d = 2^{m-r}$

Some Properties of $\text{RM}(m, r)$

- ▶ Code Parameters:
 - ▶ Blocklength, $n = 2^m$
 - ▶ Dimension, $k = \sum_{i=0}^r \binom{m}{i}$
 - ▶ Minimum distance, $d = 2^{m-r}$
- ▶ Nesting property: $\text{RM}(m, 0) \subset \text{RM}(m, 1) \subset \text{RM}(m, 2) \subset \cdots \subset \text{RM}(m, m-1) \subset \text{RM}(m, m)$
- ▶ $\text{RM}(m, r)^\perp = \text{RM}(m, m-r-1)$

Some Properties of $\text{RM}(m, r)$

- ▶ Code Parameters:
 - ▶ Blocklength, $n = 2^m$
 - ▶ Dimension, $k = \sum_{i=0}^r \binom{m}{i}$
 - ▶ Minimum distance, $d = 2^{m-r}$
- ▶ Nesting property: $\text{RM}(m, 0) \subset \text{RM}(m, 1) \subset \text{RM}(m, 2) \subset \cdots \subset \text{RM}(m, m-1) \subset \text{RM}(m, m)$
- ▶ $\text{RM}(m, r)^\perp = \text{RM}(m, m-r-1)$
- ▶ RM codes achieve Shannon capacity
[Kudekar-Kumar-Mondelli-Pfister-Sasoglu-Urbanke (2017),
Reeves-Pfister (2023), Abbe-Sandon (2023)]

Families of Algebraic Error-Correcting Codes

- ▶ BCH and Reed-Solomon codes

$$H_{\text{RS}} = \begin{bmatrix} 1 & \alpha^b & \dots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \dots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+n-k-1} & \dots & \alpha^{(n-1)(b+n-k-1)} \end{bmatrix}$$

- ▶ Reed-Muller codes

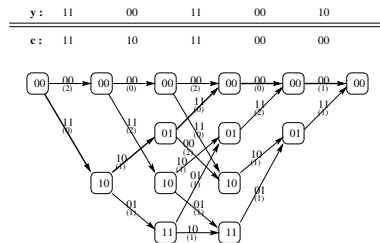
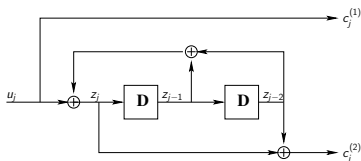
$$\text{RM}(r, m) = \{[f(\mathbf{b}(0)) \ f(\mathbf{b}(1)) \ \dots \ f(\mathbf{b}(2^m - 1))] : f \in P_r^m\},$$

where P_r^m is the set of Boolean polynomials in m variables with degree at most r .

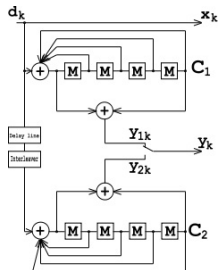
- ▶ Algebraic geometry codes, such as Goppa codes.

Sparse Graph Codes

► Convolutional codes



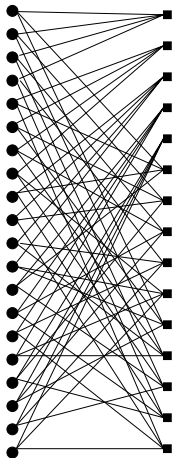
► Turbo codes



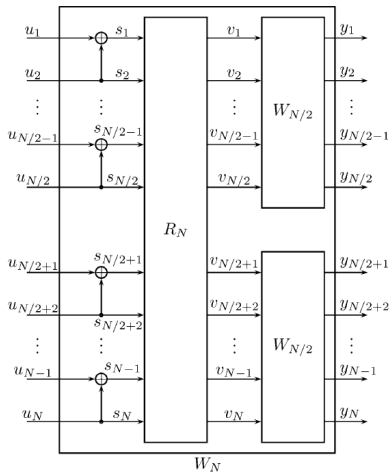
[Figure courtesy of Wikipedia]

Sparse Graph Codes

- Low-Density Parity-Check (LDPC) Codes



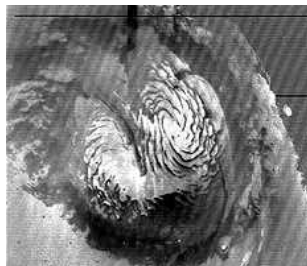
Polar Codes



[Figure taken from Arikan (2009)]

Applications of Error-Correcting Codes

Deep-Space Communications

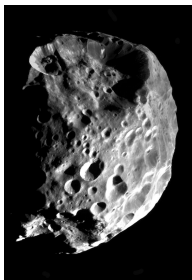


The North polar cap of Mars, taken by Mariner 9 in 1972. (Source: NASA.)

The NASA Mariner probes (1969–1973) used a powerful Reed-Muller code capable of correcting 7 errors out of 32 bits transmitted, consisting of 6 data bits and 26 parity check bits.

Applications of Error-Correcting Codes

Deep-Space Communications



Phoebe, a moon of Saturn, as imaged by the Cassini probe, 11 June 2004.
(Source: NASA.)

The Cassini orbiter to Saturn uses a **concatenated code** consisting of an outer **(255,223) Reed-Solomon code** and an inner **rate-1/6 convolutional code**. It provides a *bit error rate* of **1 in a million**.

Applications of Error-Correcting Codes

Wireless Communications



Convolutional codes, turbo codes and LDPC codes have been incorporated into various wireless communications standards.

Applications of Error-Correcting Codes

Compact Discs



To guard against scratches, cracks and similar damage, CD's use **Cross-Interleaved Reed-Solomon Coding (CIRC)**, which involves a $(28,24)$ Reed-Solomon code and a $(32,28)$ Reed-Solomon code, separated by a 28-way convolutional interleaver.

Can **correct error bursts** of up to **4000 bits** ($\sim 2.5\text{mm}$ of track).

Quantum Error-Correcting Codes



image credit: Google

- ▶ Quantum computers are far more susceptible to errors than classical digital devices.
- ▶ Quantum algorithms make essential use of **entangled** quantum states.
- ▶ Entangled states are extremely fragile, and **decohere** quickly.
- ▶ So, error correction has to be an integral component of any quantum computer.

Bibliography

- [1] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.
- [2] W.C Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge Univ. Press, 2003.
- [3] S. Lin and D. Costello, *Error Control Coding*, 2nd ed., Pearson, 2004.
- [4] R.M. Roth, *Introduction to Coding Theory*, Cambridge Univ. Press, 2006.
- [5] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge Univ. Press, 2008.