# LECTURE 1
# A SHORT INTRODUCTION TO CLASSICAL ERROR-CORRECTING CODES

Advanced Quantum Information and Computing

Thomas Debris-Alazard

Inria, École Polytechnique

*Building an efficient quantum computer?*

Let's go $\big($good luck. . . $\big)$! But it is impossible to build architectures that are completely isolated from the environment: decoherence $\big($pure states $\mapsto$ mixed states$\big)$

**Decoherence ($\longleftrightarrow$ Quantum Noise):**

There will be "noise" during computations that will modify the results. . .

- What does the "noise" mean?
- How to be "protected" against the "noise"?

$\longrightarrow$ Do the classical computation also suffer of errors during computations?

*Building an efficient quantum computer?*

Let's go $\big($good luck...$\big)$! But it is impossible to build architectures that are completely isolated from the environment: decoherence $\big($pure states $\mapsto$ mixed states$\big)$

**Decoherence ($\longleftrightarrow$ Quantum Noise):**

There will be "noise" during computations that will modify the results...

- What does the "noise" mean?
- How to be "protected" against the "noise"?

$\longrightarrow$ Do the classical computation also suffer of errors during computations?

Yes!

How do we proceed to be protected against errors in classical computations?

In the early age: errors in computation, big issue!

$\longrightarrow$ Read the story of R. Hamming in the Bell labs $\left(1947\right)$:

https://en.wikipedia.org/wiki/Richard_Hamming

**Classically:**

- Resource that we need to protect: **the bits** 0 and 1
- Errors: for instance bits are **flipped** $\begin{cases} 0 \mapsto 1 \\ 1 \mapsto 0 \end{cases}$

If communications $\left(\text{with bits}\right)$ are not efficient, do we only need to **improve physical devices**?

$\longrightarrow$ Information theory and coding theory offer **an alternative** $\left(\text{and much more exciting}\right)$!

In the early age: errors in computation, big issue!

$\longrightarrow$ Read the story of R. Hamming in the Bell labs $\left(1947\right)$:

https://en.wikipedia.org/wiki/Richard_Hamming

**Classically:**

- Resource that we need to protect: the bits 0 and 1
- Errors: for instance bits are flipped $\begin{cases} 0 \mapsto 1 \\ 1 \mapsto 0 \end{cases}$

Breakthrough: Shannon $\left(1948/1949\right)$ gave the foundations to protect classical computations against errors but not only!

Protection against errors in computation $\subsetneq$ Information theory

Protect against errors in the quantum world: a much harder problem!

- **Problem 1:** Not enough to protect $|0\rangle$ and $|1\rangle$, every linear combinations $\alpha |0\rangle + \beta |1\rangle$ must be protected as well
- **Problem 2:** Much richer error model than for classical bits $\big($not only "flip"...$\big)$
- **Problem 3:** Impossibility to copy qubits before working on it $\big($no cloning theorem$\big)$
- **Problem 4:** Measurements modify the qubits...

To overcome these issues: take a look on how we proceed in the classical case!

A short introduction to classical error-correcting codes!

$\longrightarrow$ There is a rich $\Big($and still extremely active$\Big)$ underlying theory!

It also turns out that classical error correcting codes appear almost everywhere

in computer science $\Big($and mathematics$\Big)$

# THE REPETITION CODE

Suppose that we send bits across a noisy channel

001011 $\rightsquigarrow$ 001111

How can the receiver detect that an error occurred and correct it?

**But also an issue for the memory:**

Suppose that we stored 001011 on a magnetic memory but after some years it has been altered and we now had 001111. How to recover the initial data?

**Do what you do in your everyday life:**

Add redundancy!

**An example: spell your name over the phone, send first names!**

M like Mike, O like Oscar, R like Romeo, A like Alpha, I like India and N like November

- We perform an encoding (*i.e.*, adding redundancy),

$$\text{M} \mapsto \text{Mike},\ \text{O} \mapsto \text{Oscar},\ \text{R} \mapsto \text{Romeo},\ \text{A} \mapsto \text{Alpha, etc.}\ldots$$

- We send the names across the noisy channel (given by a bad communication over the phone),

$$\text{Mike} \xrightarrow{\text{noise}} \text{“ike”},\ \text{Oscar} \xrightarrow{\text{noise}} \text{“scar”},\ \text{Romeo} \xrightarrow{\text{noise}} \text{“meo”},\ \text{Alpha} \xrightarrow{\text{noise}} \text{“ alph”}$$

- The receiver can perform a decoding: recovering the first names and then the letters,

$$\text{“ike”} \to \text{Mike} \to \text{M},\ \text{“sca”} \to \text{Oscar} \to \text{O},\ \text{“meo”} \to \text{Romeo} \to \text{R},\ \text{“alph”} \to \text{Alpha} \to \text{A}$$

$$\text{To transmit } \mathbf{m} \in \{0,1\}^k \xrightarrow{\text{(encoding)}} \mathbf{c} \in \{0,1\}^n \xrightarrow[\text{channel}]{\text{noisy}} \mathbf{y} = \mathbf{c} + \mathbf{e}$$

Aim: recover **m** from **y**!

**Important remark:**

We mapped $k$ to $n > k$ bits $\left(\text{redundancy}\right)$: $\mathbf{c}$ encoding of $\mathbf{m}$

*The set of encoding* $\mathbf{c} \in \{0,1\}^n$ *for* $\mathbf{m} \in \{0,1\}^k$ *is called a* ***code***

**Decoding phase:**

Recover $\mathbf{m}$ from $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c}$ is the encoding of $\mathbf{m}$

**Your first (error correcting) code: 3-repetition code**

Encoding 1 bit into 3 bits,

$$\begin{aligned} 0 &\longmapsto 000 \\ 1 &\longmapsto 111 \end{aligned}$$

$\{(000, 111)\}$ is called the three repetition code!

**Exercise:**

Suppose that errors can only be bit flipping $(0 \mapsto 1 \text{ and } 1 \mapsto 0)$. What does it mean to successfully remove an error with the above encoding? Which errors can you successfully remove?

- **Encoding**: $b \in \{0,1\} \longmapsto bbb \in \{0,1\}^3$
- **Noisy Channel**: $bbb \longmapsto c_1c_2c_3$ where $p \stackrel{\text{def}}{=} \mathbb{P}(c_i \neq b)$
- **Decoding Strategy**: given $c_1c_2c_3 \in \{0,1\}^3$, choose the majority bit

$$001 \longmapsto 0,\ 011 \longmapsto 1,\ 101 \longmapsto 1, \text{ etc.} \ldots$$

$\longrightarrow$ This decoding strategy is successful if there are $< 2$ errors

| Successful Decoding with probability | Unsuccessful Decoding with probability |
| --- | --- |
| $(1-p)^3 + 3p(1-p)^2$ | $p^3 + 3(1-p)p^2$ |

Suppose that $p = 0.01$,

- The decoding procedure fails for the 3 repetition code with probability $3 \times 10^{-4}$
- The same decoding procedure with the 5 repetition code fails with probability $\approx 10^{-5}$

Which code will you use for communication?

# CODE RATE

prob. successfully decoding 5-repetition code $\gg$ prob. successfully decoding 3-repetition code

But. . .

prob. successfully decoding 5-repetition code $\gg$ prob. successfully decoding 3-repetition code

But...

Encoding 1 bit necessitates $5 > 3$ bits!

$\longrightarrow$ Higher communication cost with the 5-repetition code...

**Code rate:**

Given an encoding from $k$ bits to $n$ bits, *i.e.* $\mathbf{m} \in \{0,1\}^k \longmapsto \mathbf{c} \in \{0,1\}^n$, its rate is defined as:

$$R \stackrel{\text{def}}{=} \frac{k}{n}$$

- The 3-repetition code has rate $1/3 = 0.33\cdots$
- The 5-repetition code has rate $1/5 = 0.2$

# CODE RATE TENDING TO $0$ TO ENSURE RELIABLE COMMUNICATION?

*Suppose that $p = 0.01$ is the probability that a bit is flipped across the noisy channel*

- The majority voting fails for the 3 repetition code with probability $3 \times 10^{-4}$
- The majority voting fails for the 5 repetition code with probability $\approx 10^{-5}$

But,

- The 3-repetition code has rate $1/3 = 0.33\cdots$
- The 5-repetition code has rate $1/5 = 0.2$

Is the rate necessarily go to 0 in order to fail the decoding phase with probability tending to 0?

No! Second Shannon's theorem

$\longrightarrow \forall$ Rate $\leq$ Channel Capacity

It is possible to decode with probability of success tending to 1!

*Shannon's noisy-channel coding theorem has* **two parts***: one positive and one negative*

**Shannon's noisy-channel coding theorem:**

1. For every channel $Q$, the channel capacity $C(Q) \in (0, 1)$ has the following property: for all $\varepsilon > 0$ and $R < C(Q)$, for large enough $n$, there exists a code $\subseteq \{0, 1\}^n$ whose rate is $\geq R$, and an associated decoding algorithm such that the probability of error is $< \varepsilon$
2. Reciprocally, for any code $\subseteq \{0, 1\}^n$ with rate $R > C(Q)$, whatever is the decoding algorithm, its probability of error will tend $\left(\text{with } n\right)$ to 1

*Informal statement*: we did not define properly what is

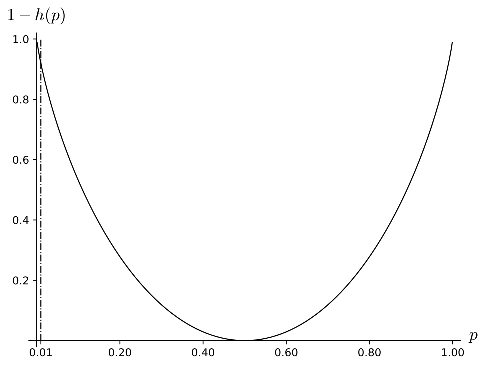
- A channel
- The capacity of a channel

$\longrightarrow$ For more details see CSC_51063_EP content $\left(\text{Lectures 6 and 7}\right)$!

$p = 0.01$: the 3-repetition code fails to decode with probability $3 \times 10^{-4}$ with a rate $0.33\ldots$

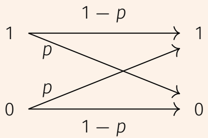
But capacity: $C(0.01) = 1 - h(0.01) = 0.919$ where $h(x) \stackrel{\text{def}}{=} -x \log_2 x - (1-x)\log_2(1-x)$

We can do much better! Even with success probability tending to 1



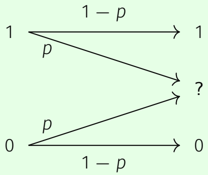
Up to now we considered the following noise model:

**Binary Symmetric Channel BSC($p$):**




$\longrightarrow$ There many other (realistic) channel models! For instance by scratching a CD-ROM you remove bits

**Exercise: Binary Erasure Channel BEC($p$)**




Is it “easier” to decode the 3-repetition repetition when BSC or BEC? What do you conclude?

# LINEAR CODES

$$\mathbf{m} \in \{0,1\}^k \xrightarrow{\text{(encoding)}} \mathbf{c} \in \{0,1\}^n$$

*A first pre-requisite to enable efficient communication over a noisy channel: we want the mappings*

$$\mathbf{m} \mapsto\in \mathbf{c} \quad \text{and} \quad \mathbf{c} \mapsto \mathbf{m}$$

*to be efficient*

**Issue:**

There are $2^k$ messages $\mathbf{m} \in \{0,1\}^k \dots$

$\longrightarrow$ It seems that we need to store a table of **exponential size** with all the mappings $(\mathbf{m}, \mathbf{c})$

To overcome this issue: **linear codes**!

$\mathbb{F}_2 = \{0, 1\}$ where

$$0 + 0 = 0,\ 1 + 0 = 0 + 1 = 1,\ 1 + 1 = 0 \quad ; \quad 1 \times 0 = 0 \times 1 = 0 \times 0 = 0,\ 1 \times 1 = 1$$

$\mathbb{F}_2^n = \underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{n \text{ times}}$ is a $\mathbb{F}_2$-vector space

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$

$$\forall \lambda \in \mathbb{F}_2, \quad \lambda \cdot (x_1, \ldots, x_n) = (\lambda \times x_1, \ldots, \lambda \times x_n)$$

Concepts as subspaces, dimensions, etc... are defined in $\mathbb{F}_2^n$

**Linear codes:**

A linear code $\mathcal{C}$ is a subspace of $\mathbb{F}_2^n$

When $\mathcal{C}$ has dimension $k$, we say that it is an $[n, k]$-code: $n$ length, $k$ dimension

**Repetition code of length 3:**

$$\left\{(0,0,0),(1,1,1)\right\} \text{ is a } [3,1]\text{-code}$$

**Exercise:**

Show that an $[n, k]$-code has size $2^k$

**$(U, U+V)$ code:**

Given two linear codes $U, V \subseteq \mathbb{F}_2^{n/2}$, $(U, U+V) \stackrel{\text{def}}{=} \left\{(\mathbf{u}, \mathbf{u}+\mathbf{v}) : \mathbf{u} \in U \text{ and } \mathbf{v} \in V\right\} \subseteq \mathbb{F}_2^n$

**Exercise Session:**

What is the dimension of the above linear code?

*How to represent an* $[n, k]$*-code? It has size* $2^k$*, is a table of this size necessary?*

*How to represent an $[n, k]$-code? It has size $2^k$, is a table of this size necessary?*

No!

**Basis/Primal representation:**

An $[n, k]$-code $\mathcal{C}$ admits a basis $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{F}_2^n$

$$\mathcal{C} = \left\{ \mathbf{m}\mathbf{G} : \mathbf{m} \in \mathbb{F}_2^k \right\} \text{ where the rows of } \mathbf{G} \in \mathbb{F}_2^{k \times n} \text{ are the } \mathbf{b}_i\text{'s}$$

The matrix $\mathbf{G}$ is called a generator matrix of $\mathcal{C}$

Given a generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ of $\mathcal{C}$ with dimension $k$,

- We can efficiently encode $\mathbf{m} \in \mathbb{F}_2^k$ as $\mathbf{mG}$ $\left(\text{multiplication matrix-vector}\right)$
- From $\mathbf{c} = \mathbf{mG} \in \mathcal{C}$ we can easily recover $\mathbf{m}$. *But how?*

Given a generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ of $\mathcal{C}$ with dimension $k$,

- We can efficiently encode $\mathbf{m} \in \mathbb{F}_2^k$ as $\mathbf{mG}$ $\big($multiplication matrix-vector$\big)$
- From $\mathbf{c} = \mathbf{mG} \in \mathcal{C}$ we can easily recover $\mathbf{m}$. *But how?*

1. By a Gaussian elimination compute $\mathbf{S} \in \mathbb{F}_2^{k \times k}$ non-singular such that $\mathbf{SG} = (\mathbf{I}_k \mid \mathbf{A})$ $\big($up to a permutation of the columns$\big)$
2. Then $\mathbf{c} = \mathbf{mS}^{-1}\mathbf{SG} = \mathbf{mS}^{-1}(\mathbf{I}_k \mid \mathbf{A})$, and $\mathbf{m} = (c_1, \dots, c_k)\mathbf{S}$

$\longrightarrow$ This is nothing else than the procedure to solve a linear system

**Conclusion:**

The encoding $\mathbf{m} \mapsto \mathbf{c}$ and $\mathbf{c} \mapsto \mathbf{m}$ are efficient procedures $\big($only linear algebra$\big)$

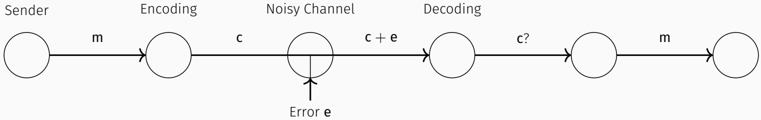
# THE BIG PICTURE

How to transmit $k$ bits over a **noisy channel**?

1. **Linear code:** fix $\mathcal{C}$ subspace $\subseteq \mathbb{F}_2^n$ of dimension $k < n$ with generator matrix $\mathbf{G}$
2. **Encoding:** map $(m_1, \ldots, m_k) \longrightarrow \mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C}$ task adding $n - k$ bits redundancy

$$\longrightarrow \text{as } \mathcal{C} \text{ is linear the encoding is easy } \Big(\text{only linear algebra}\Big), \textit{ i.e. } \mathbf{c} = \mathbf{mG}$$

3. Send $\mathbf{c}$ across the noisy channel, errors happen and some bits of $\mathbf{c}$ are modified




**Decoding:**

$\longrightarrow$ from $\mathbf{c} + \mathbf{e}$: recover $\mathbf{e}$ and then $\mathbf{c}$. Now as $\mathbf{G}$ has rank $k$, we easily recover $\mathbf{m}$

by Gaussian elimination $\Big($we use the linearity$\Big)$

# DUAL REPRESENTATION

*Linear codes as subspaces can also be written as the kernel of a matrix*

**Dual code:**

Given an $[n,k]_q$-code $\mathcal{C}$, its dual $\mathcal{C}^{\perp}$ is an $[n,n-k]$-code defined as

$$\mathcal{C}^{\perp} = \left\{ \mathbf{c}^{\perp} \in \mathbb{F}_2^n : \ \forall \mathbf{c} \in \mathcal{C}, \ \langle \mathbf{c}^{\perp}, \mathbf{c} \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{n} \underbrace{c_i^{\perp} c_i}_{\in \mathbb{F}_2} = 0 \right\}$$

**Parity-check/Dual Representation:**

$\mathcal{C}^{\perp}$ is an $[n,n-k]$-code. Furthermore, for any generator matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ (rows of $\mathbf{H}$ form a basis of $\mathcal{C}^{\perp}$) we have,

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathbb{F}_2^n : \ \mathbf{H}\mathbf{c}^{\mathsf{T}} = \mathbf{0} \right\}$$

Such matrix $\mathbf{H}$ is called a parity-check matrix of $\mathcal{C}$

**Exercise: from one representation to the other:**

From a parity-check matrix we can efficiently compute a generator matrix and reciprocally (basically only Gaussian elimination)

**Proof:**

It is clear that $\mathcal{C}^{\perp}$ is a subspace of $\mathbb{F}_2^n$. Let us show that $\mathcal{C}^{\perp}$ has dimension $n-k$. First, $\mathcal{C}$ can be written as the right kernel of a matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ with rank $n-k$,

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathbb{F}_2^n : \ \mathbf{H}\mathbf{c}^{\mathsf{T}} = \mathbf{0} \right\}$$

Therefore, all rows of $\mathbf{H}$ are elements in $\mathcal{C}^{\perp}$ showing that $\dim \mathcal{C}^{\perp} \geq n-k$. On the other hand, if $\mathbf{B} \in \mathbb{F}_2^{m\times n}$ is a basis $\Big($considering its rows$\Big)$ of $\mathcal{C}^{\perp}$. Then by linearity $\mathcal{C}$ is included in the $\Big($right$\Big)$ kernel of $\mathbf{B}$. We deduce that $k = \dim \mathcal{C} \leq n - \dim \mathcal{C}^{\perp}$ concluding the whole proof

To transmit $\mathbf{m} \in \{0,1\}^k \xrightarrow{\text{(encoding)}} \mathbf{c} \in \{0,1\}^n \xrightarrow[\text{channel}]{\text{noisy}} \mathbf{y} = \mathbf{c} + \mathbf{e}$

Aim: recover **m** from **y**!

It is equivalent to recover **c** or **e**

**A fundamental computation:**

Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$ and $\mathbf{H}$ parity-check matrix of $\mathcal{C}$:

$$\mathbf{H}\mathbf{y}^{\intercal} = \mathbf{H}(\mathbf{c} + \mathbf{e})^{\intercal} = \mathbf{H}\mathbf{c}^{\intercal} + \mathbf{H}\mathbf{e}^{\intercal} = \mathbf{H}\mathbf{e}^{\intercal}$$

$\longrightarrow$ We used the fact that $\mathbf{c} \in \mathcal{C}$ and therefore by definition $\mathbf{H}\mathbf{c}^{\intercal} = \mathbf{0}$

$\mathbf{H}$ *enables to extract information about* $\mathbf{e}$ *from* $\mathbf{y} = \mathbf{c} + \mathbf{e}$

Let $\mathcal{C}_{\mathrm{Ham}}$ be the $[7,4]$-code with parity-check matrix:

$$\mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Let $\mathbf{c}+\mathbf{e}$ where $\begin{cases} \mathbf{c} \in \mathcal{C}_{\mathrm{Ham}} \\ \text{only one bit of } \mathbf{e} \text{ is } 1 \end{cases}$ : how to easily recover $\mathbf{e}$?

Let $\mathcal{C}_{\text{Ham}}$ be the $[7,4]$-code with parity-check matrix:

$$\mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Let $\mathbf{c} + \mathbf{e}$ where $\begin{cases} \mathbf{c} \in \mathcal{C}_{\text{Ham}} \\ \text{only one bit of } \mathbf{e} \text{ is } 1 \end{cases}$ : how to easily recover $\mathbf{e}$?

1. Compute:
$$\mathbf{H}(\mathbf{c}+\mathbf{e})^{\intercal} = \mathbf{H}\mathbf{c}^{\intercal} + \mathbf{H}\mathbf{e}^{\intercal} = \mathbf{H}\mathbf{e}^{\intercal}$$
2. $\mathbf{e}$ has only one non-zero bit, $\mathbf{H}\mathbf{e}^{\intercal}$ is a column of $\mathbf{H}$
3. Columns of $\mathbf{H}$ are the binary representation of $1, 2, \ldots, 7$: $\mathbf{H}\mathbf{e}^{\intercal}$ gives (in binary) the position where there is an error!

Hamming codes can correct one error!

$\longrightarrow$ There are more clever codes than repetition or Hamming codes... In particular these codes don't seem "good". We will see later a criteria (minimum distance) for "good codes"

# A QUICK REMINDER: QUOTIENT SPACES

Given two finite subspaces: $F \subseteq E$

Equivalence relation: $x \sim y \iff x - y \in F$

$$E/F = \{\bar{x} \ : \ x \in E\} \quad \text{where} \quad \bar{x} \overset{\text{def}}{=} \{y \in E \ : \ x \sim y\} = x + F$$

$\longrightarrow$ It defines a linear space!

$$\textbf{dim}\, E/F = \textbf{dim}\, E - \textbf{dim}\, F$$

Rough analogy:

| $E/F$ | $\mathbb{Z}/4\mathbb{Z}$ |
|---|---|
| $\{\overline{x_1}, \dots, \overline{x_N}\}$ | $\{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$ |
| $\overline{x_i} = x_i + F$ | $\bar{\ell} = \ell + 4\mathbb{Z}$ |
| $\bar{x} = \bar{y} \iff x - y \in F$ | $\bar{\ell} = \overline{m} \iff \ell - m \in 4\mathbb{Z}$ |
| $E = \bigsqcup_{1 \leq i \leq N} \overline{x_i}$ | $\mathbb{Z} = \bigsqcup_{\ell \in \{0,1,2,3\}} \bar{\ell}$ |

Decoding: given $\mathbf{c} + \mathbf{e}$, recover $\mathbf{e}$

$\longrightarrow$ Make modulo $\mathcal{C}$ to extract the information about $\mathbf{e}$

**Coset space: $\mathbb{F}_2^n/\mathcal{C}$**

$$\sharp\, \mathbb{F}_2^n/\mathcal{C} = 2^{n-k} \quad \text{and} \quad \mathbb{F}_2^n/\mathcal{C} = \left\{ \overline{\mathbf{x}}_i \ : 1 \leq i \leq 2^{n-k} \right\} = \left\{ \mathbf{x}_i + \mathcal{C} \ : \ 1 \leq i \leq 2^{n-k} \right\}$$

where the $\mathbf{x}_i$'s are the representatives of $\mathbb{F}_2^n/\mathcal{C}$. The $x_i + \mathcal{C}$'s are disjoint!

A natural set of representatives via a parity-check $\mathbf{H}$: syndromes

**Proposition:**

We have:

1. $\mathbf{x}_i + \mathcal{C} \in \mathbb{F}_2^n/\mathcal{C} \longmapsto \mathbf{H}\mathbf{x}_i^{\mathsf{T}} \in \mathbb{F}_2^{n-k}$ $\Big($called a syndrome$\Big)$ is an isomorphism
2. $\mathbb{F}_2^n = \bigsqcup_{\mathbf{s} \in \mathbb{F}_2^{n-k}} \left\{ \mathbf{z} \in \mathbb{F}_2^n \ : \ \mathbf{H}\mathbf{z}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}} \right\}$

$$\mathbf{c} + \mathbf{e} \bmod \mathcal{C} = \mathbf{H}(\mathbf{c} + \mathbf{e})^{\mathsf{T}} = \underbrace{\mathbf{H}\mathbf{c}^{\mathsf{T}}}_{=0} + \mathbf{H}\mathbf{e}^{\mathsf{T}} = \mathbf{H}\mathbf{e}^{\mathsf{T}} \text{ which gives information to recover } \mathbf{e} \ \Big(\text{decoding}\Big)$$

$\longrightarrow$ $\mathbf{c} + \mathbf{e} \bmod \mathcal{C}$ is only function of $\mathbf{e}$!

**Proof:**

1. Let us first show that $\mathbf{x}_i + \mathcal{C} \in \mathbb{F}_q^n/\mathcal{C} \longmapsto \mathbf{H}\mathbf{x}_i^{\mathsf{T}} \in \mathbb{F}_2^{n-k}$ is a well-defined mapping. If we choose another class representative $\mathbf{y}_i + \mathcal{C} = \mathbf{x}_i + \mathcal{C}$. Then by definition

$$\mathbf{y}_i - \mathbf{x}_i \in \mathcal{C} \iff \mathbf{H}(\mathbf{y}_i - \mathbf{x}_i)^{\mathsf{T}} = 0 \iff \mathbf{H}\mathbf{y}_i^{\mathsf{T}} = \mathbf{H}\mathbf{x}_i^{\mathsf{T}}$$

   It shows that we have a well-defined mapping. But the equivalence also shows that it is a one-to-one mapping

   The above application is surjective as $\mathbf{H}$ has rank $n-k$, therefore for any $\mathbf{s} \in \mathbb{F}_2^{n-k}$ it exists $\mathbf{x} \in \mathbb{F}_2^n$ such that $\mathbf{H}\mathbf{x}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}}$ and $\mathbf{x}$ defines one representative. Furthermore the mapping is clearly linear, concluding the proof of 1

2. This is a consequence of the equivalence relation but let's give a direct proof. We have shown above that $\forall \mathbf{z} \in \mathbb{F}_2^n$, it exists $\mathbf{s} \in \mathbb{F}_2^n$ such that $\mathbf{H}\mathbf{z}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}}$ $\left(\mathbf{H} \text{ has rank } n-k\right)$.

   To conclude notice that $\left\{\mathbf{z} \in \mathbb{F}_2^n \;:\; \mathbf{H}\mathbf{z}^{\mathsf{T}} = \mathbf{s}^{\mathsf{T}}\right\}$ are clearly disjoint for $\mathbf{s} \in \mathbb{F}_2^{n-k}$

$\mathcal{C}$ be an $[n, k]$-code with generator and parity-check matrices $\mathbf{G}$ and $\mathbf{H}$

- Given a noisy codeword, $\mathbf{y} = \underbrace{\mathbf{c}}_{\in \mathcal{C}} + \mathbf{e}$, its syndrome is

$$\mathbf{H}\mathbf{y}^{\intercal} = \mathbf{H}\mathbf{c}^{\intercal} + \mathbf{H}\mathbf{e}^{\intercal} = \mathbf{H}\mathbf{e}^{\intercal} \text{ where we used } \mathcal{C} = \left\{\mathbf{c} \in \mathbb{F}_2^n : \ \mathbf{H}\mathbf{c}^{\intercal} = \mathbf{0}\right\}$$

- Given a syndrome, $\mathbf{s}^{\intercal} = \mathbf{H}\mathbf{e}^{\intercal}$, we can easily compute its associated noisy codeword, by a Gaussian elimination we compute $\mathbf{y}$ such that $\mathbf{H}\mathbf{y}^{\intercal} = \mathbf{s}^{\intercal}$ $\left(\text{as } \mathrm{rank}(\mathbf{H}) = n - k\right)$

$$\mathbf{H}\mathbf{y}^{\intercal} = \mathbf{s}^{\intercal} \iff \mathbf{H}(\mathbf{y} - \mathbf{e})^{\intercal} = \mathbf{0} \iff \mathbf{y} - \mathbf{e} \in \mathcal{C} \iff \mathbf{y} = \underbrace{\mathbf{c}}_{\in \mathcal{C}} + \mathbf{e}$$

# HAMMING AND MINIMUM DISTANCES

**Hamming weight:**

$$\forall \mathbf{x} \in \mathbb{F}_2^n, \quad |\mathbf{x}| \stackrel{\text{def}}{=} \sharp\left\{ i \in [1,n], \ x_i \neq 0 \right\}$$

**Hamming Distance:**

$$d_{\mathrm{H}}(\mathbf{x},\mathbf{y}) \stackrel{\text{def}}{=} \sharp\left\{ i \in [1,n] : \ x_i \neq y_i \right\}$$

$$\longrightarrow \quad d_{\mathrm{H}}(\mathbf{x},\mathbf{y}) = |\mathbf{x}-\mathbf{y}|$$

**Remark:**

Be careful: $|\cdot|$ is not a norm but $d_{\mathrm{H}}(\cdot,\cdot)$ is a distance

An important parameter for a code: its minimum distance

$\longrightarrow$ It measures the quality of a code in terms of "error detection"

**Minimum Distance:**

Given $\mathcal{C} \subseteq \mathbb{F}_2^n$, its minimum distance is defined as

$$d_{\min}(\mathcal{C}) \stackrel{\text{def}}{=} \min \left\{ |\mathbf{c}_1 - \mathbf{c}_2| : \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \text{ and } \mathbf{c}_1 \neq \mathbf{c}_2 \right\}$$

**Remark:**

For a linear code $\mathcal{C}$,

$$d_{\min}(\mathcal{C}) = \min \left\{ |\mathbf{c}| : \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\} \right\}$$

Suppose that someone sends us a codeword $\mathbf{c} \in \mathcal{C}$ across a noisy channel

Our goal is to guess if an error occurred

*How can we proceed? What is the maximal amount of errors for which we can take the right decision with certainty?*

Suppose that someone sends us a codeword $\mathbf{c} \in \mathcal{C}$ across a noisy channel

Our goal is to guess if an error occurred

*How can we proceed? What is the maximal amount of errors for which we can take the right decision with certainty?*

**Error detection strategy:**

Given a received $\mathbf{y}$ we compute $\mathbf{H}\mathbf{y}^\top$ for $\mathbf{H}$ being a parity-check matrix of the code. If we obtain $\mathbf{0}$ then we say that no error occurred

This strategy gives the right answer with certainty if the Hamming weight of the error is $< d_{\min}(\mathcal{C})$!

**Proof:**

If an error occurred then we receive $\mathbf{c} + \mathbf{e}$. Therefore $\mathbf{H}\left(\mathbf{c} + \mathbf{e}\right)^\top = \mathbf{H}\mathbf{c}^\top + \mathbf{H}\mathbf{e}^\top = \mathbf{H}\mathbf{e}^\top$. Then if $|\mathbf{e}| < d_{\min}(\mathcal{C})$ we necessarily have $\mathbf{e} \notin \mathcal{C}$ and $\mathbf{H}\mathbf{e}^\top \neq \mathbf{0}$. However, if $|\mathbf{e}| \geq d_{\min}(\mathcal{C})$ it is possible that $\mathbf{e} \in \mathcal{C}$ and $\mathbf{H}\mathbf{e}^\top = \mathbf{0}$

*If the Hamming weight of the error is* $< d_{\mathrm{min}}(\mathcal{C})$ *we can detect it*

Is there some kind of such criteria over the error to ensure that we can successfully decode?

*If the Hamming weight of the error is $< d_{\min}(\mathcal{C})$ we can detect it*

Is there some kind of such criteria over the error to ensure that we can successfully decode?

$\longrightarrow$ Yes!

**A decoding strategy:**

Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$, compute

$$\mathbf{c}_0 \in \mathcal{C} \text{ such that } |\mathbf{y} - \mathbf{c}_0| = \min \left( |\mathbf{y} - \mathbf{c}_1| : \mathbf{c}_1 \in \mathcal{C} \right)$$

**If $|\mathbf{e}| < d_{\min}(\mathcal{C})/2$, then $\mathbf{c}_0 = \mathbf{c}$ and our decoding is successful!**

$$\mathbf{x} \in \mathbb{F}_2^n, \quad \mathcal{B}(\mathbf{x}, r) \stackrel{\text{def}}{=} \left\{ \mathbf{y} \in \mathbb{F}_2^n \ : \ |\mathbf{y} - \mathbf{x}| \leq r \right\}$$

**Proposition:**

Given a code $\mathcal{C} \subseteq \mathbb{F}_2^n$,

$$\forall \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \quad \mathbf{c}_1 \neq \mathbf{c}_2: \qquad \mathcal{B}\left(\mathbf{c}_1, \left\lfloor \frac{d_{\min}(\mathcal{C})-1}{2} \right\rfloor\right) \bigcap \mathcal{B}\left(\mathbf{c}_2, \left\lfloor \frac{d_{\min}(\mathcal{C})-1}{2} \right\rfloor\right) = \emptyset$$

**Proof:**

By contradiction, suppose there exists $\mathbf{y} \in \mathcal{B}\left(\mathbf{c}_1, \left\lfloor \frac{d_{\min}(\mathcal{C})-1}{2} \right\rfloor\right) \bigcap \mathcal{B}\left(\mathbf{c}_2, \left\lfloor \frac{d_{\min}(\mathcal{C})-1}{2} \right\rfloor\right)$,

$$\begin{aligned} |\mathbf{c}_1 - \mathbf{c}_2| &= |(\mathbf{c}_1 - \mathbf{y}) - (\mathbf{c}_2 - \mathbf{y})| \\ &\leq |\mathbf{c}_1 - \mathbf{y}| + |\mathbf{c}_2 - \mathbf{y}| \qquad \text{(triangular inequality)} \\ &\leq \left\lfloor \frac{d_{\min}(\mathcal{C}) - 1}{2} \right\rfloor + \left\lfloor \frac{d_{\min}(\mathcal{C}) - 1}{2} \right\rfloor \\ &< d_{\min}(\mathcal{C}) \end{aligned}$$

which is a contradiction as $\mathbf{c}_1 \neq \mathbf{c}_2$ and they belong to $\mathcal{C}$ with minimum distance $d_{\min}(\mathcal{C})$

When transmitting $\mathbf{c} \in \mathcal{C}$, if the Hamming weight of the error is $< d_{\min}(\mathcal{C})/2$, then computing the closest codeword for the Hamming distance necessarily gives $\mathbf{c}$

**Proposition:**

Given a linear code $\mathcal{C}$ with parity-check matrix $\mathbf{H}$, the $\mathbf{He}^{\mathsf{T}}$ are distinct when $|\mathbf{e}| < d_{\text{min}}(\mathcal{C})/2$

**Proof:**

See Exercise Session

# BE CAREFUL

When transmitting $\mathbf{c} \in \mathcal{C}$, if the Hamming weight of the error is $< d_{\min}(\mathcal{C})/2$, then computing the closest codeword for the Hamming distance necessarily gives $\mathbf{c}$

The above statement says that with $< d_{\min}(\mathcal{C})/2$ errors the decoder computing the closest codeword for the Hamming distance succeeds **with certainty**!

$\longrightarrow$ There are codes for which computing the closest codeword works with probability $1 - \mathrm{e}^{-Cn}$ as soon as there are $\leq d_{\min}(\mathcal{C})$ errors, we gain a factor two!

(in particular random codes, for more details see Lecture 8 in CSC_51063_EP)

# CONCLUSION

- Adding redundancy, a process called encoding, enables to be protected against errors
- Shannon's theorem: not too much redundancy needs to be added to be protected against the noise $\Big($via the capacity of the noisy channel$\Big)$
- Linear codes are nice objects to be able to perform efficiently the encoding
- In practice: consider the noise as flipping the bits

  $\longrightarrow$ But it is not the only model of noise
- Hamming weight enables to quantify the amount of errors
- The minimum distance is a good quantity to quantity the amount of noise which can be decoded and detected

**Conceptual hard part of the lecture:**

Familiarize yourself with the coset point of view $\Big($via syndromes$\Big)$

# BE CAREFUL ABOUT THE DECODING PHASE

**About Shannon's theorem**

Given a noisy channel $Q$, Shannon tells us that it exists a code which can be decoded if and only if its rates is $< C(Q)$ $\Big($capacity of the channel$\Big)$

$\longrightarrow$ It does not explicit a code with an associated efficient decoding algorithm!

**About the closest codeword:**

Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$, computing the closest codeword is a hard task $\Big($we don't know how to efficiently perform this operation$\Big)$

*It turns out that designing codes with an efficient decoding algorithm is a very hard task! It is still an active research topic with deep implications in practice*

Few families of codes with an efficient decoding algorithm are known. For instance:

- Reed-Solomon codes and the family of Algebraic Geometric $\Big($AG$\Big)$ codes
- Polar codes derived from $(U, U+V)$-codes
- Convolutional codes

- See lectures (and exercise sessions) from CSC_51063_EP
- Nice lecture notes by Alain Couvreur (with a focus on algebra):

  http://www.lix.polytechnique.fr/~alain.couvreur/doc_ens/lecture_notes.pdf
- The “bible” of error correcting codes: *The theory of error correcting codes*, F.J. MacWilliams , N.J.A. Sloane (1978)

Error correcting codes have a huge impact in theoretical computer science, cryptography, communications, quantum key distribution (QKD), etc. . .

The approach given in this lecture is at the core of the design of quantum error correcting codes

# EXERCISE SESSION