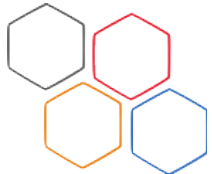


LIRMM

Cours de codes 1 sur 3

Eleonora Guerrini

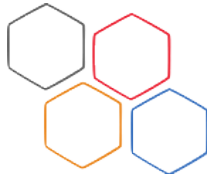




A code: What for ?

- Recover faulty transmitted data
- Distributed Data Storage
- Conceive Fault Tolerant Algorithms

(ALGO DISTRIBUÉE)



A code: what is it ?

Definition

Un code correcteur est un ensemble de vecteurs (mots) et un couple d'algorithmes (Enc, Dec) qui gèrent la transmission des mots sur un canal bruité.

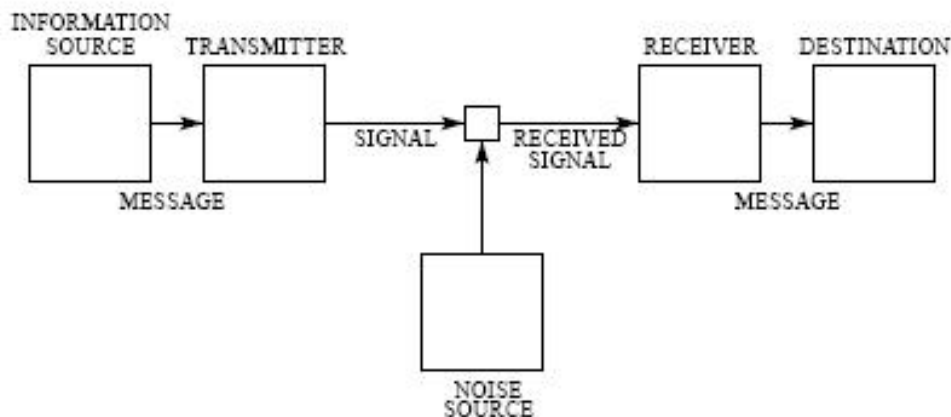
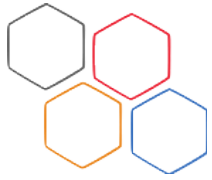
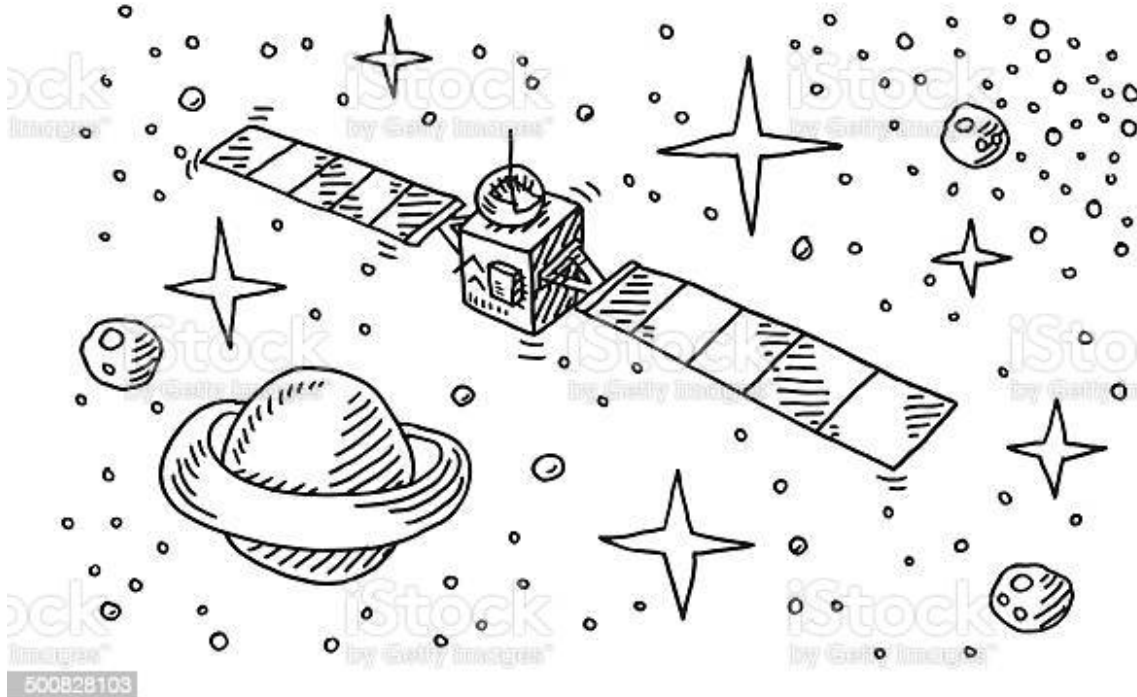


Fig. 1—Schematic diagram of a general communication system.

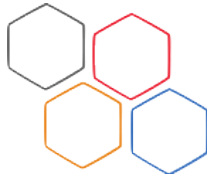


Codes correcteurs pour les transmissions

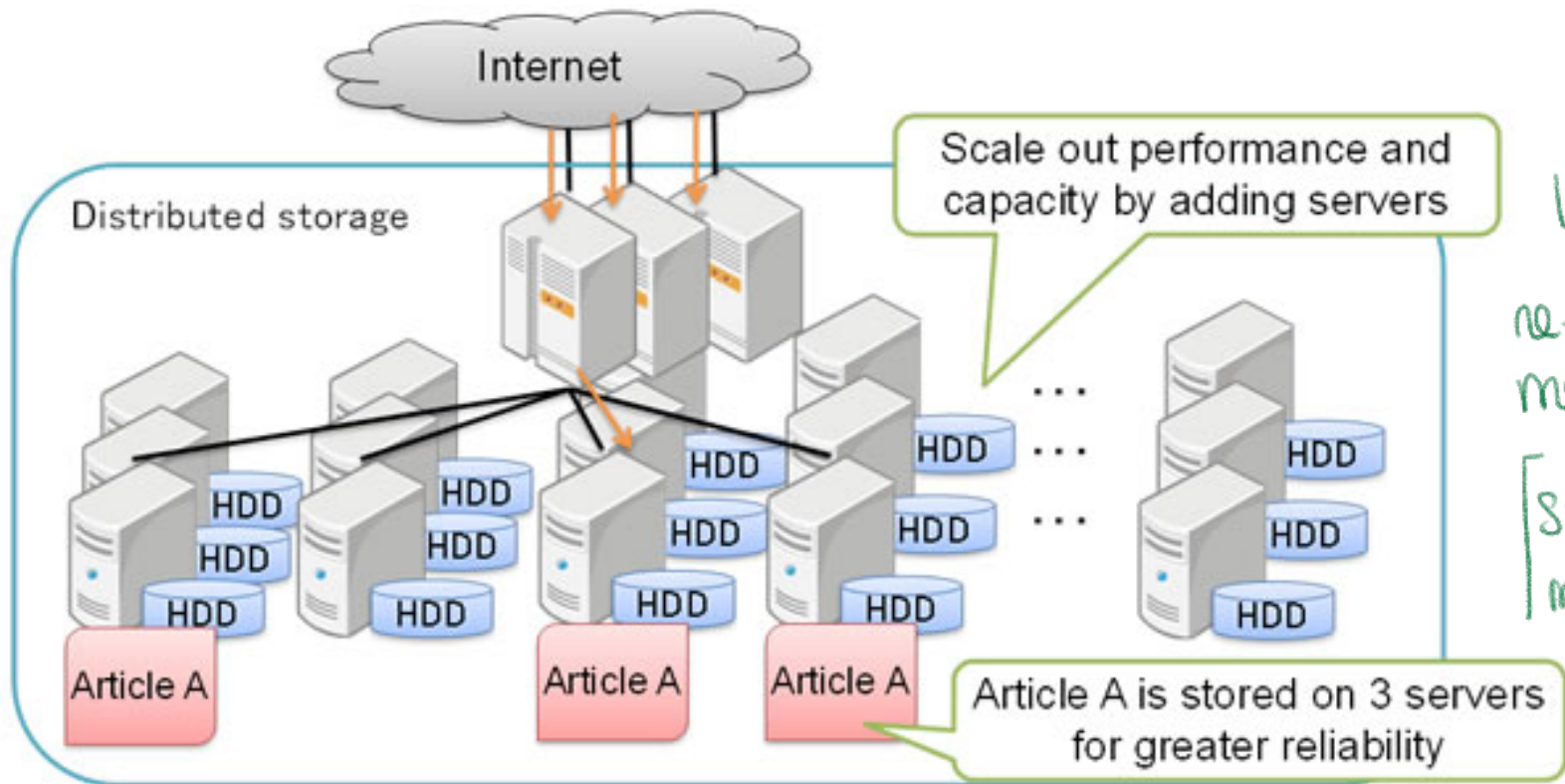


Dans les cas des transmissions
couleuses, il n'est pas
possible de re-tromettre
le message en cas de
perturbation (ex. Satellites)

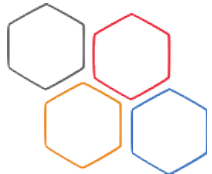
⊕ Besoin de corriger même
si couleux



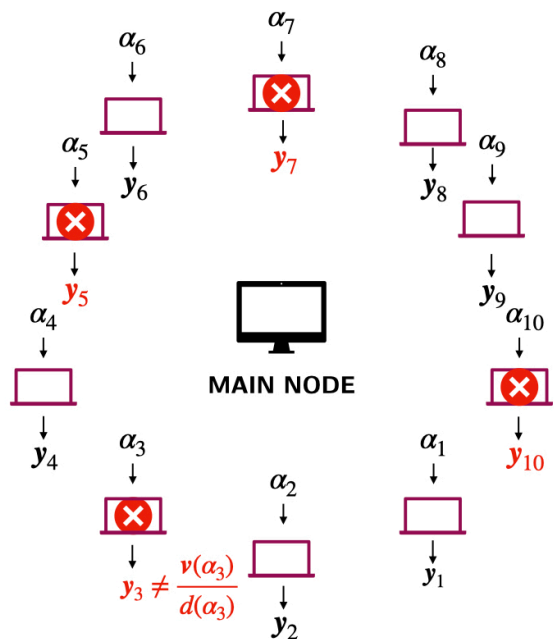
Codes correcteurs dans le stockage d'information



Impossible de re-demander le message endommagé
[si un serveur de stockage meurt, il faut avoir la possibilité de récupérer la donnée autrement]



Codes correcteurs pour les algorithmes tolérants aux fautes

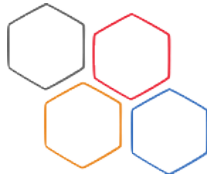


- cas algo distribuée

- Nœud defectueux

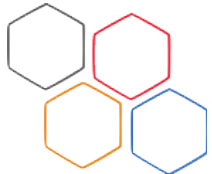
- Nœud malveillant

[ex: Produit matriciel en parallèle]



Les questions fondamentales du cours 1

- Modèle: le code, le canal, le bruit
- Algorithmes: Encoder et decoder
- Parametres :Distance d'un code et Théorème de Shannon
- Exemples :code d'Hamming et decodage



Modèle de Shannon: le code, le canal, le bruit

Enc : Encodeur
 Dec : Decodeur

$m \in (\mathbb{F}_2)^k \longrightarrow c \in (\mathbb{F}_2)^n$
 Hamming 4 $\xrightarrow{k=4}$ $n=7$

Bruit: Additif

$y_{reçu} \exists \tilde{c} \in (\mathbb{F}_2)^n$ tq

$y = c + e$

ex $c = (0, 0, 0, 0, 0, 0, 0)$
 $y = (1, 0, 0, 0, 0, 0, 0)$
 $e = (1, 0, 0, 0, 0, 0, 0)$

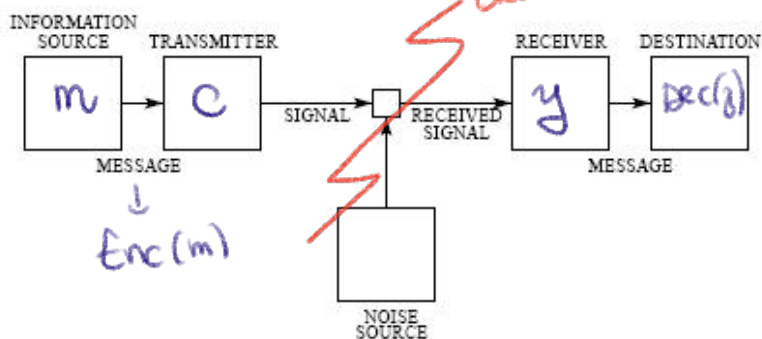
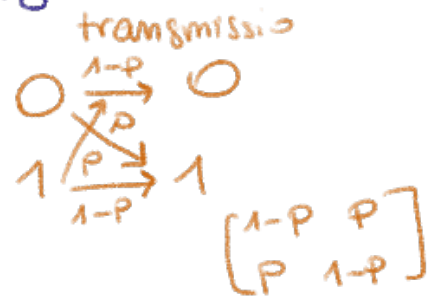


Fig. 1—Schematic diagram of a general communication system.

$Dec(y) = Enc(m)$



MATRICE de transition

- BSC (Binary Symmetric Channel)

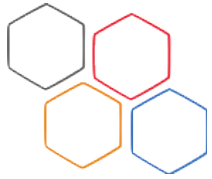
CANAL :

- Sans mémoire

$P_r(y_i \neq c_i)$ est indep de $P_r(y_{i-1} \neq c_{i-1})$
 autrement $P_r(y_i | c_i)$ est indep de $P_r(y_{i-1} \neq c_{i-1})$

- P : Proba que il y a un flip de bit $0 < p < 1/2$ (sinon on inverse)

- Symétrique $P_r(c_i \text{ flip de } 0 \rightarrow 1) = P_r(c_i \text{ flip de } 1 \rightarrow 0)$



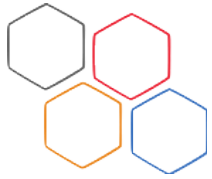
Encoding and Decoding

Si \cos q-aire $0 \rightarrow 0$
 $1 \xrightarrow{1-p} \vdots$
 $\vdots \xrightarrow{p} \vdots$
 $q-1 \xrightarrow{q-2} q-1$

Enc(m) : $(\mathbb{F}_2)^k \rightarrow (\mathbb{F}_2)^n$
 $\{0,1\}^k \rightarrow \{0,1\}^n \quad n > k \text{ redondance}$
 - Injective
 $e \in (\mathbb{F}_2)^n$

dec(m) et Enc(m) on le choisit linéaire pour "efficacité"

Enc(m) linéaire \Rightarrow Algo d'algèbre linéaire pour encoder m
 $\hookrightarrow C$ est un sous-esp. vect. de $(\mathbb{F}_2)^n$ de dim k



Theorème de Shannon

Théorie de la Codes correcteur
"GRAAL de la Théorie de la Codes correcteur"

p loi de proba (binaire)
 $H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$

Formulation "simple" du theore

- tout canal est bruité

CAPACITE' (max inform. qu'on peut transmettre de façon fiable)

- $C_{sp} = 1 - H(p)$ (correction)
- R taux de transmission (redundance)

dim esp mess \rightarrow k rendement du code
 \rightarrow n longueur code (rate)



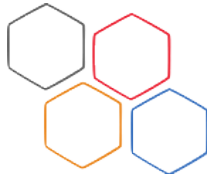
code Hamming
 $n=7$
 $k=4 \rightarrow \frac{4}{7} ?$
corrigé 1 err.

th Shannon

C_{sp} fixé = $1 - H(p)$, p proba d'erreurs du canal Pour envoyé et y reçu.

$0 < p < 1/2$
 $\varepsilon \leq 1/2 - p$
Pour $n \gg 0$

1. $\exists \delta > 0$ (Enc, Dec)
si $k \leq \lfloor (1 - H(p) + \varepsilon) \cdot n \rfloor$ alors
 $Pr(Dec(y_{reçu}) \neq c) \leq 2^{-\delta n}$



Definition Code et Parametres

2. Pour $k > \lceil (1 - H(p) + \epsilon)n \rceil$ pour tout (ϵ, δ)

$$\exists c \in \{0,1\}^n \text{ tq } P_n(\text{dec}(y) \neq c) > 1/2$$

- Soit Σ un alphabet, $(\Sigma)^k$ l'espace des messages et k et n des entiers naturels tels que $k \leq n$.

$$E_n(m) : (\mathbb{F}_2)^k \rightarrow (\mathbb{F}_2)^n$$

- $C \subset (\Sigma)^n$:

- Linearité: $E_n(0) = 0$

$$E_n C(m_1 + m_2) = E_n C(m_1) + E_n C(m_2)$$

- On appelle k la dimension du code

$\rightarrow E_n C((\mathbb{F}_2)^k)$ est esp. vect de $(\mathbb{F}_2)^n$

- Rate: k/n

$$x = (x_1, \dots, x_n) \quad y = (y_1, \dots, y_n)$$

$$d_H(x, y) = \sum (x_i + y_i)$$

- Rapp. avec une base, G induit base

- Distance (?) Hamming : $d_H(x, z) = 0$

$$d_H(x, y) = d_H(y, z)$$

$$E_n C(m) = m \cdot G$$

$$d_H(x, y) + d_H(y, z) \geq d_H(x, z) \quad \text{Hamming } k=4 \quad (\mathbb{F}_2)^4$$

$$mG = (1, 0, 1, 1, 0, 1, 0) \in (\mathbb{F}_2)^7$$

mot de code de Hamming

$$x = (0, 1, 0, 1)$$

$$m = (1, 0, 1, 1); \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$d_H(x, 0) = 2$$



$$y = (1, 0, 0)$$
$$z = (1, 0, 1)$$

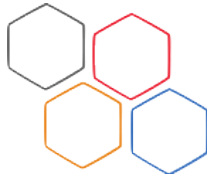
(preuve vue)

• Correction Detection Effacements

Distance (d'Hamming) d'un code C

$$\min_{x_1 \neq x_2} (d(x_1, x_2) \mid x_1, x_2 \in C)$$

Quantifions les erreurs avec la distance et la définition du code



Detection et Correction: Modèle de Hamming

Philosophie de décodage MLD
 Maximum Likelihood Decoding (maximum de vraisemblance)

Si y reçue, on veut $\text{Dec}(y) = c$

Problématique et Algorithmes: Rôle de la distance

$$P_r \{y \text{ reçue} | c \text{ envoyée}\} = \prod_{i=1}^n P_r \{y_i \text{ reçue} | c_i \text{ envoyée}\}$$

$P_r(y \text{ reçue} | c \text{ envoyée})$ plus grande possible

$$n - d(y, c) = \#\{i | y_i = c_i\}$$

$$(1-p)^{n-d(y,c)} \cdot p^{d(y,c)}$$

\leftarrow si $y_i = c_i$
 \leftarrow $y_i \neq c_i$

$$c = \begin{matrix} 1 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \end{matrix}$$

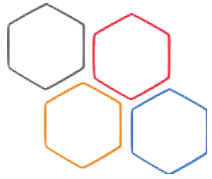
$$y = \begin{matrix} 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow \end{matrix}$$

$d(y, c)$

Algorithme de décodage naïf : MLD Algorithme

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

plus grand possible \Rightarrow
 $d_H(y, c)$ plus petite possible



MLD decoder : bons et mauvais cotés

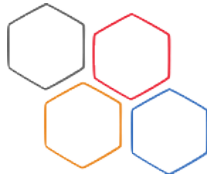
MLD Problème NP-hard

In: $y \in \mathbb{F}_2^n$, \mathcal{C} code ($\mathcal{C} \subseteq \mathbb{F}_2^n$ de dim k)

Out: $c \in \mathcal{C}$ tq $d(y, c) = \min\{d(y, c) \mid c \in \mathcal{C}\}$

(même si code lin)

On se restreint à un cas plus précis où on met une borne sur ce qu'on peut corriger



Modèles de Décodage

ex si $\bar{c} = (1001100)$
 $e = (1100000)$
 $y = (0101100)$

$$y = \bar{c} + e$$

e : vecteur err-
 \bar{c} : mot envoyé

BDD Decoder

Bounded Decoding Distance

In: y reçu, \mathcal{C} code, t borne sur les erreurs qu'on veut (peut-être?) corriger

Out: e tq $d_H(c, y) \leq t$

si \mathcal{C} est Code Hamming

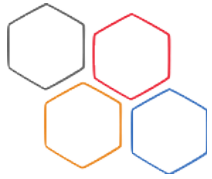
si $t = 2$

Le decodeur nous donnera un e tq $d_H(c, y) \leq 2$
 mais pas forcément $e = \bar{c}$
 r envoyé

BMD Decoder

t est le "bon" codé que il existe trs un seul mot de code possible en Output

— FIN COURS 1 —



Code de répétition et Code de parité

Hanning $R = \frac{4}{7}$ corr = 1 err

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

Repet: $\frac{3}{9}$ corr = 1 err.

- Code de répétition :

Peut détecter 2 erreurs
(3 pas possib. si une erreur par bloc qui se repete)

$$m \in (\mathbb{F}_2)^k \xrightarrow{\text{Enc}(m)} c = (m, m, m) \in (\mathbb{F}_2)^{3k}$$

ex: $(100) \xrightarrow{\text{Enc}(m)} (100 \ 100 \ 100)$

$$\begin{cases} y \neq c & \text{si } e \neq 0 \\ y = c & \text{si } \vec{e} = 0 \end{cases}$$

$(111 \ 100 \ 100)$

ici je peux détecter 2 erreurs, et donc ce cas spécifique aussi les corriger par vote de majorité mais en générale on ne peut pas corriger 2 err.

- Codes de parité :

$$m \in (\mathbb{F}_2)^k \xrightarrow{\text{Enc}(m)} c = (m, \oplus m_i) \in (\mathbb{F}_2)^{k+1}$$

ex: $(100) \xrightarrow{\text{Enc}(m)} c = (100 \ 1)$

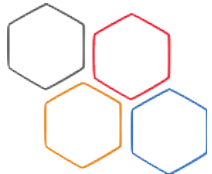
$R = \frac{k}{k+1}$ super bien

Détecte 1 erreur ? oui car si 1 erreur seulement la parité change.

$y = (10 \ 1 \ 1)$

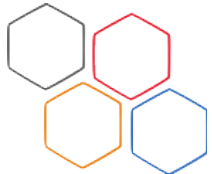
mais corrige rien

ex $(110 \ 1 \ 10 \ 100)$



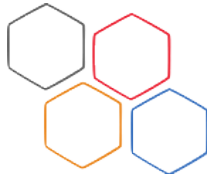
Ex : Code de repetition et parity check

- dimension du code
- longueur du code
- rendement du code : ratio k/n



Décodeurs





Exo

- Code de repetition peut corriger 1 erreur
- Code de parité peut detecter un nombre impair d'erreurs (donc au pire cas 1)

3¹-repet

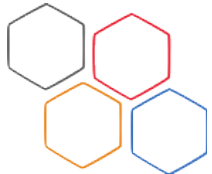
corrige 1 erreur
detecte 2 erreurs

$$R = \frac{k}{n} = \frac{1}{3}$$

parité

corrige 0
detecte 1 erreur

$$R = \frac{k}{k+1}$$



Distance et Correction: Exo

Given a code C de longueur n et dimension k , les assertions suivantes sont équivalents

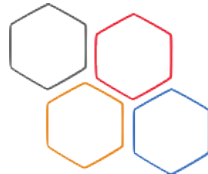
1. C a distance minimale $d \geq 2$,
2. si d est ~~impair~~, C peut corriger $\lfloor \frac{(d-1)}{2} \rfloor$ errors.
3. C peut detecter $d - 1$ errors.
4. C peut corriger $d - 1$ effacements

① \Rightarrow ③ Si $y = c + \vec{e}$ ou $d_H(c, y) = d$, comme $d(C) = d$ il peut exister c_1 t.a. $c_1 = y$. Donc Detect(y) \Rightarrow Pas erreur (car \exists mot de C)

Si $d-1$ erreur c.à.d. $d(y, c) \leq d-1$ donc y pas dans le code

① \Leftrightarrow ④ Si $d-1$ effacements ou plus \Rightarrow Unicité de c qui complète $y_i = \begin{cases} ? \\ c_i \end{cases}$ Si d effacements il peut y avoir 2 mots de code qui corresp.

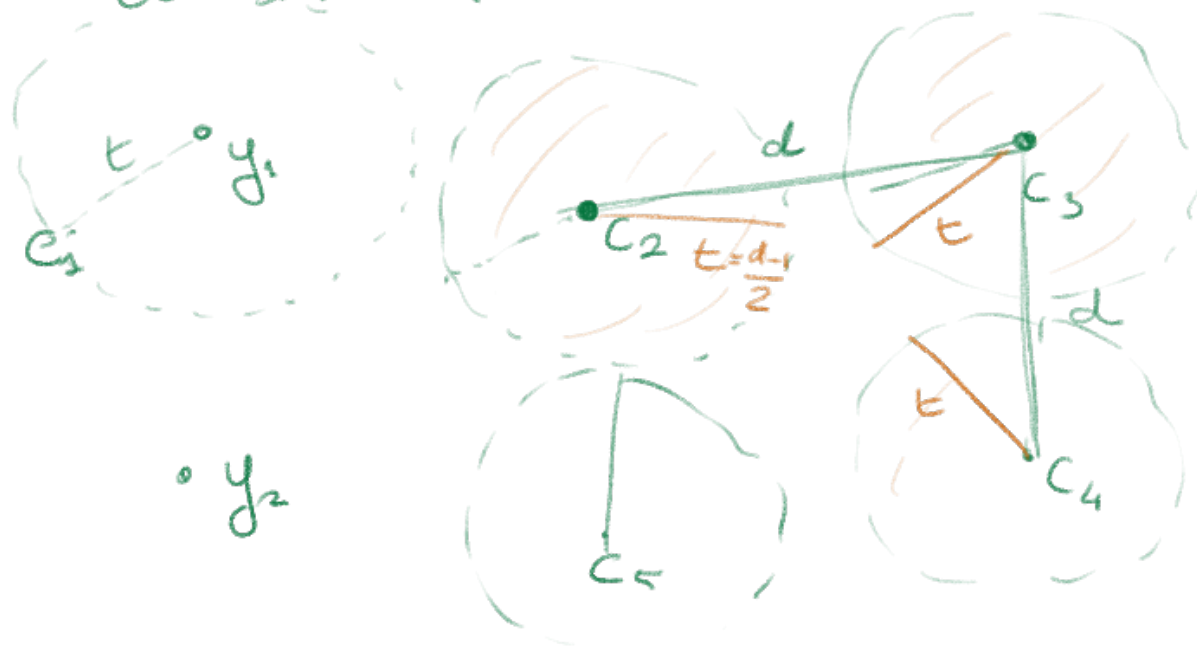
$\Leftrightarrow C$ peut corriger $d-1$ eff.



Preuve : ① \Rightarrow ② ① \Rightarrow ③ ① \Rightarrow ④ et enfin ② \Rightarrow ①
 On fait que les bouts intéressants

or ① \Rightarrow ②.

② Dire que C corrige t erreurs : \exists unique $c \in C$ tq $d_H(y, c) \leq t$ pour y reçu et t erreurs faites. Algorithme déterministe



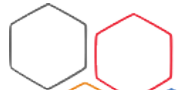
comme $d(C) = d$ alors
 tout couple de mots de code c_1, c_2 est
 tq $d_H(c_1, c_2) \geq d$

Si $t = \frac{d-1}{2}$ si $t = \frac{d-1}{2}$

$\mathcal{B}_t(c_1) \cap \mathcal{B}_t(c_2) = \emptyset$

donc si y a t erreurs
 \Downarrow

$\exists \bar{c}$ tq $d_H(c, y) \leq t \Rightarrow$
 $\exists ! \bar{c}$ tq $y \in \mathcal{B}_t(\bar{c})$



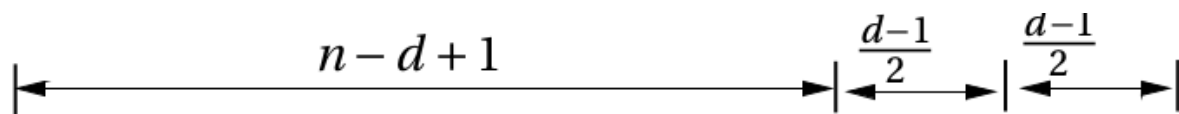
$$\textcircled{2} = D(\textcircled{1})$$

Si decode à

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

distance de \mathcal{C}

est $\geq d$



si $\exists c_1, c_2$ tq

$$c_1 \quad d_H(c_1, y) \leq t$$

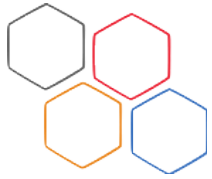
et

$$d_H(c_2, y) \leq t$$

$$c_2 \quad d_H(c_1, c_2) \leq d-1$$

y

Figure 1.3: Bad example for unique decoding.



Qu'est-ce qu'on peut espérer comme rate et distance

- Ⓐ rate (rendement) : $\frac{k}{n}$ plus grand possible
- Ⓑ t plus grand possible
- Ⓒ Corriger et détecter efficacement
- Ⓓ Calculer la distance efficacement

Qu'est-ce que c'est un bon code ?

Ⓐ + Ⓑ + Ⓒ mais Ⓐ vs Ⓑ vs Ⓒ

Hammring Ⓐ et Ⓒ mais pas de tout Ⓑ

Code Repet Ⓑ et Ⓒ mais pas de tout Ⓐ

On définit un "bon code"
celui qui a les meilleurs
paramètres possible



Comment les param-
ètres se calculent.



Borne (d'Hamming)

$$\begin{aligned} \textcircled{m} &\rightarrow \text{Enc}(m) \\ \in (\mathbb{F}_2)^k &\rightarrow c \in (\mathbb{F}_2)^n \end{aligned}$$

Boue de empilement de sphère

Pour un code $C \Rightarrow (n, k, d)$ binaire
ou $t = \frac{d-1}{2}$

$$2^k \sum_{i=0}^t \binom{n}{i} \leq 2^n$$

Si $\mathcal{B}_t(c) = \{y \in \mathbb{F}_2^n \mid d_H(c, y) \leq t\}$, ou



$$\mathcal{B}_t(c_1) \cap \mathcal{B}_t(c_2) = \emptyset \quad \forall c_1, c_2 \in C$$

Donc $\bigcup_{c \in C} \mathcal{B}_t(c) = \bigsqcup_{c \in C} \mathcal{B}_t(c)$ (# c à d toutes les intersection \emptyset)

$$\left| \bigsqcup_{c \in C} \mathcal{B}_t(c) \right| = \underbrace{\text{nb. boules}}_{\substack{\leftarrow 2^k \\ \text{nb mots de codes}}} \times \underbrace{\text{Volume d'une boule}}_{\substack{\sum_{i=0}^t \binom{n}{i} \\ \text{invariant par } c \text{ choisi, donc ici } c = \vec{0}}} \leq \underbrace{2^n}_{\substack{\text{mots possible} \\ \text{dans l'espace} \\ (\mathbb{F}_2)^n}}$$



Hamming

Pour Hamming $7+1=8$ $n=7, k=4, t=1$
on a egale

Les codes qui atteignent la borne d'empil- de spheres (c.à.d.) s'appellent PARFAIT. (Hamming est donc un code parfait)

→ Conséquence : $\forall y$ recue \exists toujours un unique c tq $d_H(y, c) = 1$ ou 0

① Comment calculer d'"efficacement" ?
la distance d'un code.

Lemme si C linéaire alors $d(C) = \min_{c \neq 0} w(c)$

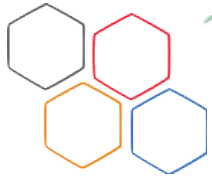
où $w(c) \stackrel{\text{def}}{=} |\{i \mid c_i \neq 0\}|$. on appelle $w(c)$ le Poids de c

ex: $(10011) \rightarrow w(c) = 3$



→ Preuve ? Il faut montrer que $c_1 = (100), c_2 = (010), c_3 = (001)$

$$\min_{\substack{c_1 \neq c_2 \\ c_1, c_2 \in C}} d_H(c_1, c_2) = \min_{\substack{c \neq 0 \\ c \in C}} w(c)$$



Pour c_1, c_2 $d(c_1, c_2) = w(c_1 - c_2)$ car \mathcal{C} est linéaire donc $c_1 - c_2 \in \mathcal{C}$

Matrice génératrice et de parité

$$\min_{c_1 \neq c_2} \{d(c_1, c_2)\} = \min_{\substack{c \in \mathcal{C} \\ c \neq 0}} \{w(c)\} + \forall w(c), c \in \mathcal{C} \\ d(\bar{c}, 0) = w(\bar{c})$$

Avantage de la linéarité :

- $d(C) = \min_{\substack{c \neq 0 \\ c \in C}} w(c)$
- C resp. vect de $(\mathbb{F}_q)^n$ de dim k

$q \in 2$

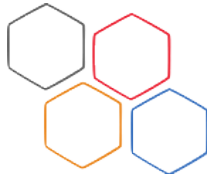
• Enc(w) : $(\mathbb{F}_2)^k \rightarrow (\mathbb{F}_2)^n$ linéaire

\Downarrow
 \exists matrice $G \in (\mathbb{F}_2)^{n \times k}$ tq

$$c = \text{Enc}(m) = G \cdot m^T$$

$$m = (m_1, \dots, m_k)$$

$$\left[\begin{array}{c} \text{ } \\ \text{ } \\ \text{ } \end{array} \right]_{n \times k} \left[\begin{array}{c} m^T \\ \text{ } \\ \text{ } \end{array} \right]_{k \times 1} = c \in (\mathbb{F}_2)^n$$



Encodage

Décodage d'un code d'Hamming

Code d'Hamming $n=7, k=4, q=2$

$$m = (x_1, x_2, x_3, x_4) \xrightarrow{\text{Enc}(m)} (x_1, x_2, x_3, x_4, x_2+x_3+x_4, x_1+x_3+x_4, x_1+x_2+x_4)$$

$C_H =$ code d'Hamming (C_H est linéaire)

$0 \in C_H$
si $c_1, c_2 \in C_H \rightarrow c_1 + c_2 \in C_H$,
 $\text{Enc}(m_1) + \text{Enc}(m_2) = \text{Enc}(m_1 + m_2) \text{ ok.}$

G matrice génératrice

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ mod } 2$$

Encodage

H matrice de parité (nature gén. du noyau)

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

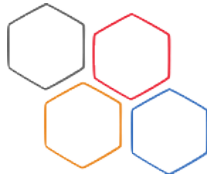
Propri. de H

$$Hy^T = \vec{0} \iff y \in C$$

$y \in (\mathbb{F}_q)^n$

Fonction d'erreur

Detect In: $y \in (\mathbb{F}_2)^n$
Out: Bool (vrai si erreurs) \rightarrow if $Hy^T \neq 0$ Bool := true



Exemple de Décodage

détection d'erreurs

Est-ce que l'on peut aussi décoder avec H?

Si \vec{y} reçu et \vec{e} erreur $\vec{y} = c + \vec{e}$ où $c \in C$

$$Hy = H(c + e) = \underbrace{Hc}_=0 + \underbrace{He}_{\text{syndrome}} \text{ vecteur de } (\mathbb{F}_2)^{n-k} \text{ qui contient l'info. des erreurs.}$$

Ex avec H :

$$c = (1000011)$$

$$He^T = 0 \Rightarrow \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \vec{0}$$

pas d'erreurs.

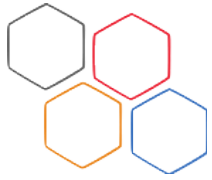
$$c = (1011010) \xrightarrow{\text{Bruit}} y = (0011010)$$

$$Hy^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \xrightarrow{\text{écriture binaire de la pos. d'err.}} (001) \rightarrow \text{Erreur à l'indice } 1$$

syndrome

$$c = (1011010)$$

- x -



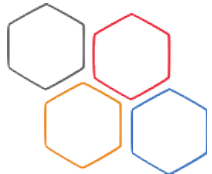
Si l'idée
Si on prend une matrice "quelconque" comme G ,
est-ce que j'obtiens un bon code ?

-
- ① Optimale par une borne
 - ② Decodage efficace NON
- [ex Hamming]
① + ②

① C'est oui grâce à la borne Gilbert-Varshamov

Construction Code : Fix n, d, \mathbb{F}_q
Greedy C_G Output $C_G \in (\mathbb{F}_q)^n$ tel $d(c) \geq d$

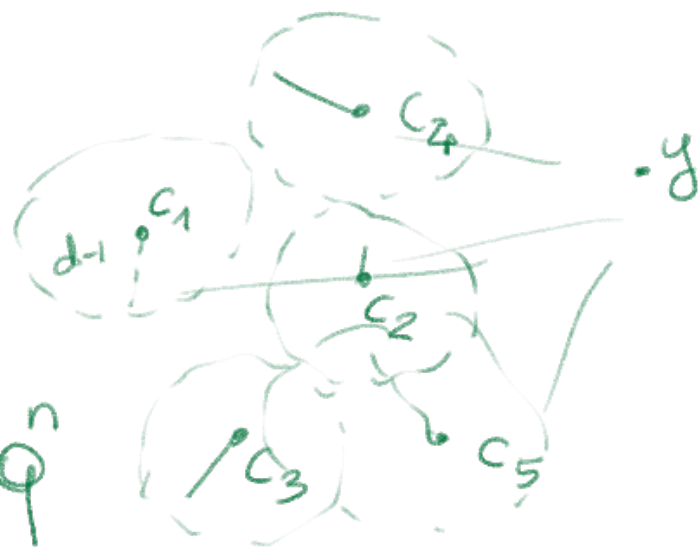
1. $C_G \leftarrow \emptyset$
2. while $\exists c \in (\mathbb{F}_q)^n$ tq $d(c, c') \geq d \forall c' \in C_G$



1. $\mathcal{C}_G = \mathcal{C}_G \cup \{0\}$
 3. Output \mathcal{C}_G

Greedy Const $(n, d) \rightarrow \mathcal{C}_G$

- ① Est-ce que $d(\mathcal{C}_G) = d$?
 - ② Est-ce que le nombre de mots dans \mathcal{C}_G est le plus grand possible (si \mathcal{C}_1 code quelconque binaire param (n, d) , alors nb de mots dans $\mathcal{C}_1 \leq$ nb mots dans \mathcal{C}_G)
- Trivial (par construction)
 → L'algo s'arrête quand $\nexists v \in (\mathbb{F}_q)^n, v \notin \mathcal{C}_G$ tq $d(v, c) \geq d \forall c \in \mathcal{C}_G$



② Après l'algo $|\cup_{c \in \mathcal{C}_G} B(c, d-1)| = q^n$

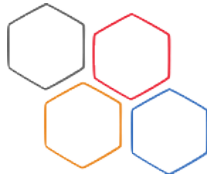
car si $e \Rightarrow y \in (\mathbb{F}_q)^n$ tq $\cup_{c \in \mathcal{C}_G} B(c, d-1) \ni y \Rightarrow d(y, c) \geq d \forall c \in \mathcal{C}_G \Rightarrow$

y pourrait être choisi dans Greedy

$\sum_{c \in \mathcal{C}_G} |B(c, d-1)| \geq |\cup_{c \in \mathcal{C}_G} B(c, d-1)| = q^n$

Borne Gilbert-Varshamov

$q^k > q^n$



$$q^k \cdot \text{Vol}_q(d-1, n) \geq q \Rightarrow \frac{1}{\text{Vol}_q(d-1, n)}$$

Attention: différence entre borne Hamming et Borne Gilbert-Varsh.

q^k : nb. de mots du code

$$q^k \leq \dots \text{Vol}_q\left(\frac{d-1}{2}, n\right) \dots$$

$$q^k \geq \dots \text{Vol}_q(d-1, n) \dots$$

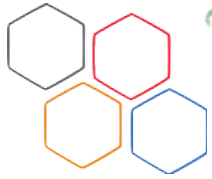
Interprétation Asymptotique -

$$\frac{k}{n} = R \quad \delta = \frac{d}{n} \quad 1 = \frac{n}{n} \Rightarrow d = \delta n$$

$$q^k \geq \frac{q^n}{\text{Vol}_q(\delta n, n)} \Rightarrow q^k \geq \frac{q^n}{q^{n H_q(\delta)}} \Rightarrow R \geq 1 - H_q(\delta)$$

Pour n, q et $\delta \in (0, 1) \exists$ famille de codes $\{t_q$ $R \geq 1 - H_q(\delta)$

? Est-ce que la borne de Gilbert-Varshanos peut être atteinte?
théorème: Un code linéaire aléatoire atteint la Borne de Gilbert-Varshanos



Preuve : Soit $w(c)$ = poids du mot c .

Il nous faut montrer que qu'il existe une matrice $k \times n$, où le rang est

$$k = (1 - H_q(\delta) - \varepsilon)n, \text{ tq}$$

$$\forall m \in (\mathbb{F}_q)^k \rightarrow w(mG) \geq d. \quad \text{(Code à distance } d)$$

$$\Pr [w(mG) < d] = \frac{|\{c \mid w(c) < d\}|}{q^n} \leftarrow \text{ mots de poids } < d$$

q^n = tout mots $\in (\mathbb{F}_q)^n$

$$d = \delta n$$

$$\Pr [\exists m \neq 0 \text{ tq } w(mG) < d] \leq \frac{q^{n H_q(\delta)}}{q^n} \leq q^{-k - \varepsilon n} \times (q^k - 1)$$

$\text{car } k \leq (1 - H_q(\delta) - \varepsilon)n$

$$\leq q^{-\varepsilon \cdot n}$$

Il nous reste à prouver que G a rang pleine.



Dire que G a rang plein equiv à montrer que $\exists m \neq 0$ tq $mG = 0$. Dans notre cas, il suffit de montrer que

$\Pr[\exists m \neq 0 \mid w(mG) = 0] \neq 0$. On réutilise la preuve précédente. \rightarrow

$$-\varepsilon_n \geq \Pr[\exists m \neq 0 \text{ tq } w(mG) < d] \geq \Pr[\exists m \neq 0 \text{ tq } w(mG) = 0]$$

