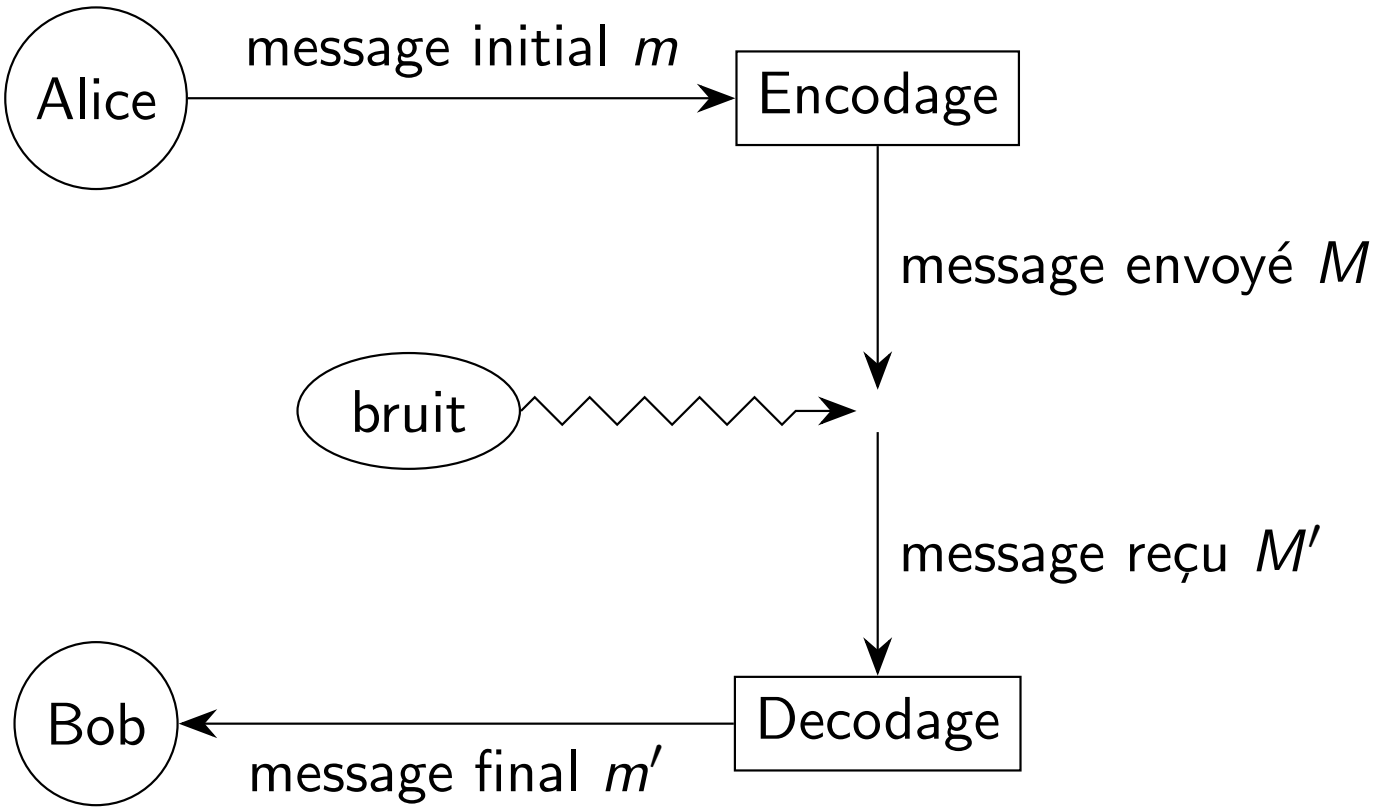
Introduction à la théorie des codes

Vanessa VITSE

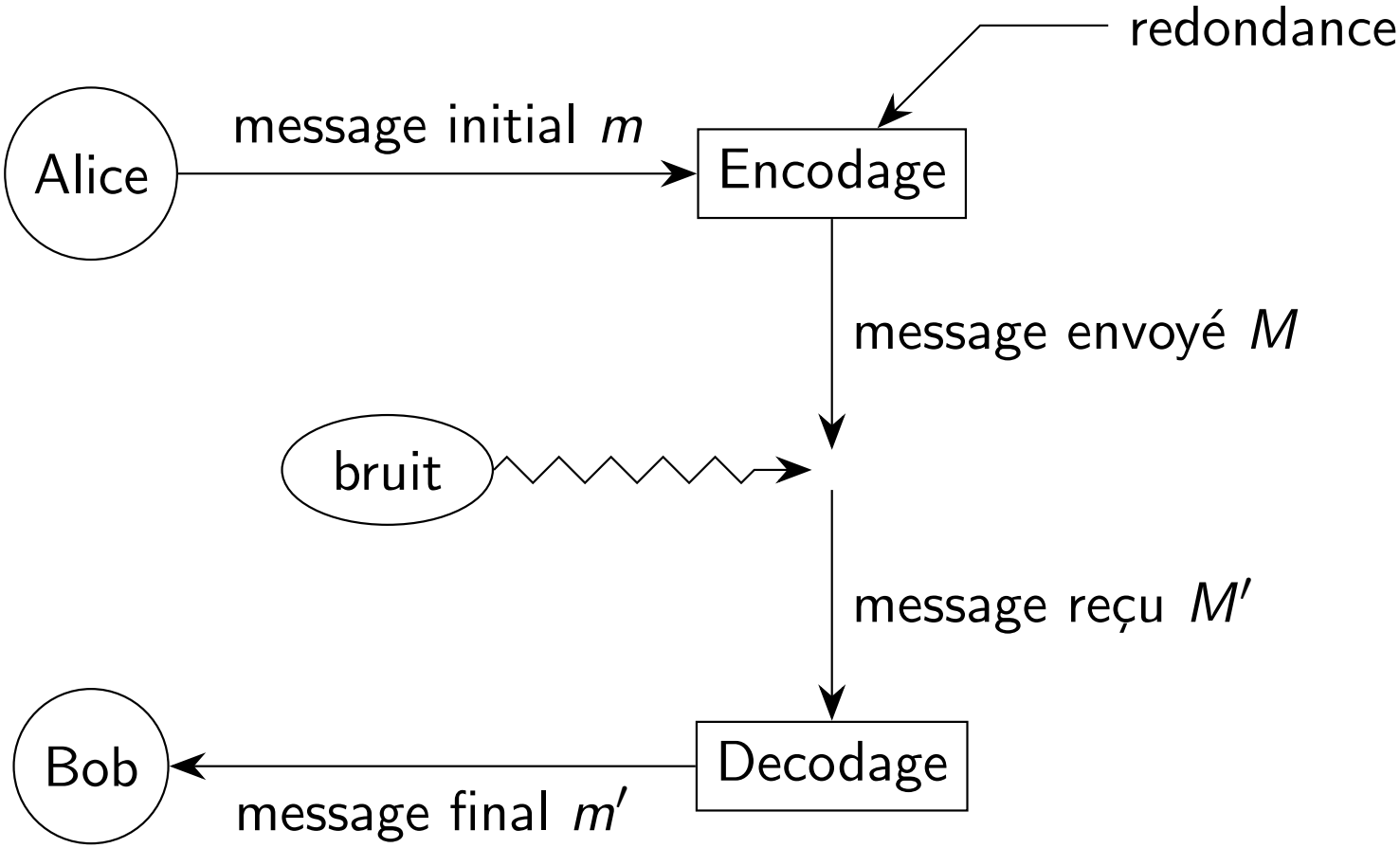
Université Grenoble Alpes

M1 Maths 2021

Transmission sur un canal



Transmission sur un canal



Section 1

Codes linéaires

Cadre des codes linéaires

Soit $\mathbb{K}$ un corps fini.

Définition

Un **code linéaire** C de taille (k, n) est un sous-espace vectoriel de dimension k de $\mathbb{K}^n$.
Un **mot du code** est un élément de C.

Dans les applications $\mathbb{K} = \mathbb{F}_2$ ou $\mathbb{F}_{2^d}$: les mots du code sont représentés comme des *séquences de bits.*

Cadre des codes linéaires

Soit $\mathbb{K}$ un corps fini.

Définition

Un **code linéaire** C de taille (k, n) est un sous-espace vectoriel de dimension k de $\mathbb{K}^n$.
Un **mot du code** est un élément de C.

Dans les applications $\mathbb{K} = \mathbb{F}_2$ ou $\mathbb{F}_{2^d}$: les mots du code sont représentés comme des *séquences de bits.*

Un **encodage** est une application linéaire injective $\phi : \mathbb{K}^k \to \mathbb{K}^n$;
son image $\mathrm{Im}(\phi)$ est un (k, n)-code.

Matrice génératrice

Convention : représentation en ligne des éléments de $\mathbb{K}^n$ et $\mathbb{K}^k$

Une **matrice génératrice** d'un (k, n)-code C est une matrice $k \times n$ dont les lignes forment une base de C.

Exemple : code par bit de parité

$C = \{w \in \mathbb{F}_2^n \mid w \text{ a un nombre pair de } 1\}$

C'est un $(n-1, n)$-code ; matrice génératrice : $\begin{pmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & 1 \end{pmatrix}$

(Prouvez-le !)

Encodage systématique

Proposition

Soit C un (k, n)-code. À permutation près des coordonnées, C a une matrice génératrice de la forme $(I_k|P)$ avec $P \in \mathcal{M}_{k,n-k}(\mathbb{K})$.

(Élimination gaussienne.) P est appelée **matrice de parité** de C.

Soit C donné par la matrice génératrice $(I_k|P)$, l'**encodage systématique** est donné par

$$\phi : \begin{array}{ccc} \mathbb{K}^k & \to & \mathbb{K}^n \\ m & \mapsto & m.(I_k|P) \end{array}$$

Ainsi $\phi(m_1 \dots m_k) = (m_1 \dots m_k \ m_{k+1} \dots m_n)$;
les $(n-k)$ symboles supplémentaires $m_{k+1}, \dots, m_n \in \mathbb{K}$ sont appelés **symboles (bits) de contrôle** et introduisent la redondance

Exercice : décrire l'encodage systématique du code par bit de parité

Syndrômes

Proposition

Soit C un (k, n)-code. Il existe une application linéaire $\sigma : \mathbb{K}^n \to \mathbb{K}^{n-k}$ telle que $C = \ker \sigma$.

On appelle $\sigma(w)$ le **syndrôme** de $w \in \mathbb{K}^n$.
Ceci permet de calculer efficacement si w est un mot du code :

$$w \in C \iff \sigma(w) = 0$$

Syndrômes

Proposition

Soit C un (k, n)-code. Il existe une application linéaire $\sigma : \mathbb{K}^n \to \mathbb{K}^{n-k}$ telle que $C = \ker \sigma$.

On appelle $\sigma(w)$ le **syndrôme** de $w \in \mathbb{K}^n$.
Ceci permet de calculer efficacement si w est un mot du code :

$$w \in C \iff \sigma(w) = 0$$

Soit C donné par sa matrice génératrice $(I_k|P)$, alors une telle application est donnée par

$$\sigma : w \mapsto w. \begin{pmatrix} -P \\ I_{n-k} \end{pmatrix} \qquad \text{(exercice : à vérifier !)}$$

La matrice $H = \begin{pmatrix} -P \\ I_{n-k} \end{pmatrix}$ est appelée matrice de **contrôle**.

Section 2

Décodage

Transmission et réception

Pour transmettre un message, il faut

- le convertir en une séquence d'éléments de $\mathbb{K}^k$ (blocs)
- pour chaque bloc m, envoyer le mot de code correspondant $w = \phi(m) \in C$

Réception :

- pour chaque bloc reçu c', vérifier si c'est un mot du code en calculant son syndrôme
- si $c' \in C$, il n'y a (probablement) pas d'erreur de transmission
- si $c' \notin C$, alors il y a erreur de transmission

Transmission et réception

Pour transmettre un message, il faut

- le convertir en une séquence d'éléments de $\mathbb{K}^k$ (blocs)
- pour chaque bloc m, envoyer le mot de code correspondant $w = \phi(m) \in C$

Réception :

- pour chaque bloc reçu c', vérifier si c'est un mot du code en calculant son syndrôme
- si $c' \in C$, il n'y a (probablement) pas d'erreur de transmission
- si $c' \notin C$, alors il y a erreur de transmission

Le but du décodage est de trouver m tel que $\phi(m)$ est le plus "proche" de c'

Distance de Hamming

Poids de Hamming et distance de Hamming

Soit $w \in \mathbb{K}^n$, son **poids de Hamming** $h(w)$ est le nombre de coordonnées non nulles de w.

La **distance de Hamming** entre deux mots $w_1, w_2 \in \mathbb{K}^n$ est $d(w_1, w_2) = h(w_1 - w_2)$.

Exemples : $h(01001100) = 3$, $d(10011000, 10010001) = 2$.

Distance de Hamming

Poids de Hamming et distance de Hamming

Soit $w \in \mathbb{K}^n$, son **poids de Hamming** $h(w)$ est le nombre de coordonnées non nulles de w.

La **distance de Hamming** entre deux mots $w_1, w_2 \in \mathbb{K}^n$ est $d(w_1, w_2) = h(w_1 - w_2)$.

Exemples : $h(01001100) = 3$, $d(10011000, 10010001) = 2$.

La distance de Hamming compte combien de symboles on doit modifier pour transformer w_1 en w_2.

Dans notre contexte : combien d'erreurs de transmission pour transformer le message envoyé w_1 en le message reçu w_2.

Distance minimale

La distance minimale d'un code linéaire

La **distance minimale** d'un code linéaire C est

$$d(C) = \min\{h(w) \mid w \in C, w \neq 0\}$$

De façon équivalente, $d(C) = \min\{d(w_1, w_2) \mid w_1, w_2 \in C, w_1 \neq w_2\}$.

Distance minimale

La distance minimale d'un code linéaire

La **distance minimale** d'un code linéaire C est

$$d(C) = \min\{h(w) \mid w \in C, w \neq 0\}$$

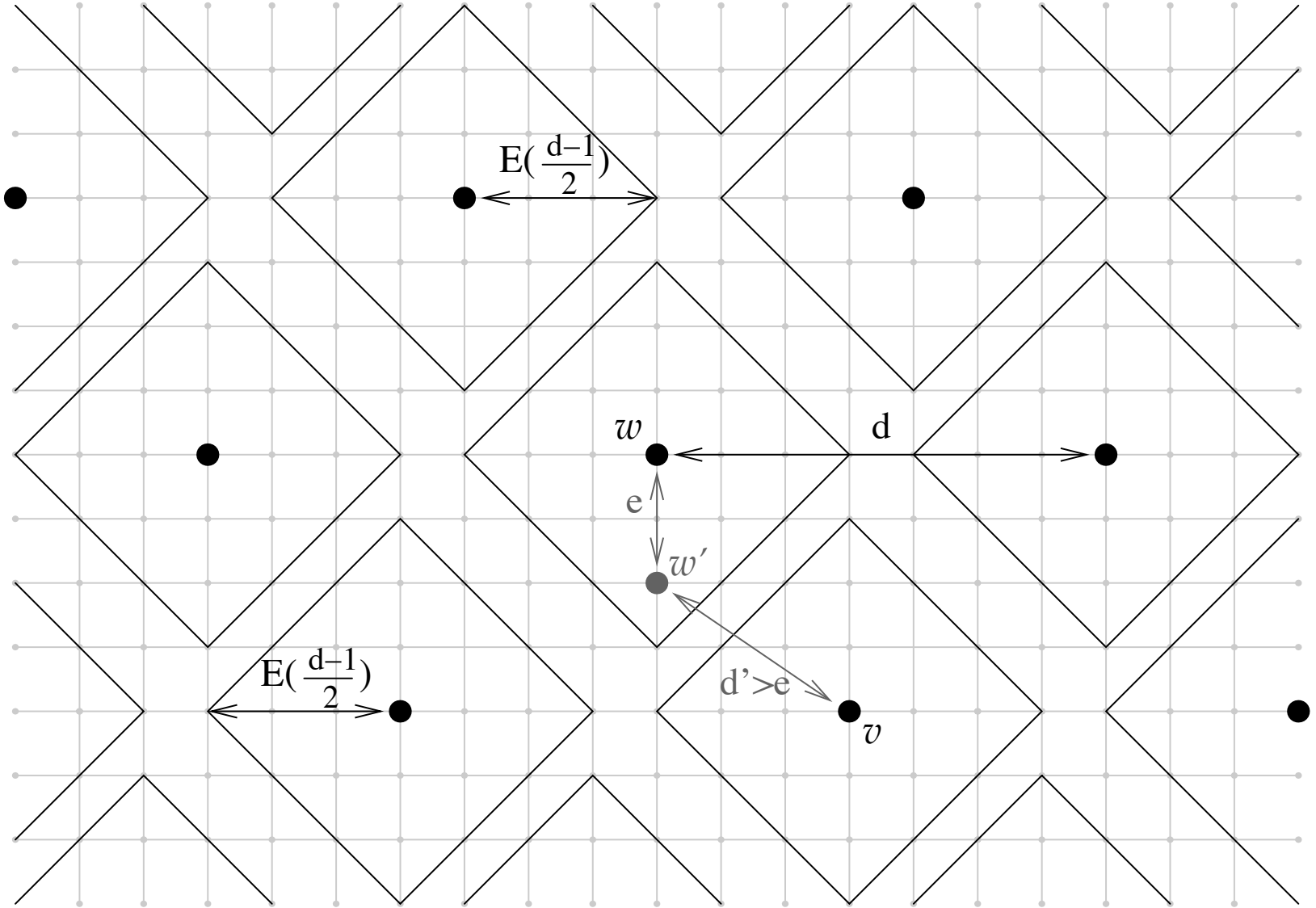
De façon équivalente, $d(C) = \min\{d(w_1, w_2) \mid w_1, w_2 \in C, w_1 \neq w_2\}$.

Théorème

Soit C un code linéaire, $w \in C$ un message envoyé, w' le message reçu, et $e = d(w, w')$ le nombre d'erreurs de transmission

- si $e < d(C)$, alors w' n'est pas un mot du code, donc les erreurs sont détectées
- si $e \leqslant \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, alors w est le mot du code le plus proche de w' (pour la distance de Hamming), donc les erreurs peuvent être corrigées.

Capacité de détection / correction



trouver le plus mot du code le plus proche w' (décoder) peut être difficile !

Décodage à l'aide du syndrôme

Détection et correction d'erreur

Étant donné un message reçu w', le but du décodage est de trouver le mot du code $w \in C$ tel que $d(w, w') = h(w' - w)$ est minimal

Soit $e = w' - w$ le **vecteur d'erreur** ; alors $\sigma(e) = \sigma(w')$
(où σ = application syndrôme).

Donc décoder $w' \iff$ trouver le plus court vecteur $e \in \mathbb{K}^n$ tel que $\sigma(e) = \sigma(w')$.

Décodage à l'aide du syndrôme

Détection et correction d'erreur

Étant donné un message reçu w', le but du décodage est de trouver le mot du code $w \in C$ tel que $d(w, w') = h(w' - w)$ est minimal

Soit $e = w' - w$ le **vecteur d'erreur** ; alors $\sigma(e) = \sigma(w')$
(où σ = application syndrôme).

Donc décoder $w' \iff$ trouver le plus court vecteur $e \in \mathbb{K}^n$ tel que $\sigma(e) = \sigma(w')$.

Il est possible de lister les vecteurs d'erreurs les plus courts correspondant à chaque syndrôme, mais uniquement si $n - k$ (et q) sont suffisamment petits.
Autrement décoder peut être très difficile

Section 3

Codes de Hamming

Bande passante versus distance minimale

Un "bon" code correcteur d'erreur

- a une grande distance minimale : capacité de corriger des grosses erreurs
- a un petit ratio n/k : meilleure gestion de la bande passante

↳ incompatibilité : un compromis doit être trouvé

Bande passante versus distance minimale

Un "bon" code correcteur d'erreur

- a une grande distance minimale : capacité de corriger des grosses erreurs
- a un petit ratio n/k : meilleure gestion de la bande passante

↳ incompatibilité : un compromis doit être trouvé

Deux cas extrêmes sur $\mathbb{F}_2$

Code par répétition : $(1, n)$-code $C_1 = \{00\ldots0,\ 11\ldots1\}$.
Code par bit de parité : $(n-1, n)$-code $C_2 = \{w \in \mathbb{F}_2^n \mid h(w) \text{ is pair}\}$.

Exercice : trouver leur distance minimale et leur capacité de correction

La borne de Hamming

Borne de Hamming

Soit un (k, n)-code C de distance minimale d sur un corps fini à q éléments,

$$\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i \leqslant q^{n-k}$$

Preuve : Il y a q^k éléments dans C, et les boules de rayon $\lfloor (d-1)/2 \rfloor$ centrées en les mots du code doivent être disjointes. Le nombre d'éléments dans de telles boules est $\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i$, donc $q^k \sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i \leqslant \#(\mathbb{F}_q^n) = q^n$.

La borne de Hamming

Borne de Hamming

Soit un (k, n)-code C de distance minimale d sur un corps fini à q éléments,

$$\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i \leqslant q^{n-k}$$

Preuve : Il y a q^k éléments dans C, et les boules de rayon $\lfloor (d-1)/2 \rfloor$ centrées en les mots du code doivent être disjointes. Le nombre d'éléments dans de telles boules est $\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i$, donc $q^k \sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i \leqslant \#(\mathbb{F}_q^n) = q^n$.

Ceci donne une borne sup sur d connaissant k et n.
Un **code parfait** est un code pour lequel l'égalité est satisfaite

Codes de Hamming

Très peu de codes parfaits : code trivial, certains codes par répétitions, codes de Hamming, et les deux cas très spéciaux des codes de Golay

Code de Hamming = code parfait avec $d = 3$

Codes de Hamming

Très peu de codes parfaits : code trivial, certains codes par répétitions, codes de Hamming, et les deux cas très spéciaux des codes de Golay

Code de Hamming = code parfait avec $d = 3$

Si C est un tel code, alors tout mot w' qui n'est pas un mot du code est à distance de Hamming 1 d'un unique mot du code
$\iff$ pour tout syndrôme non nul $s \in \mathbb{K}^{n-k}$, il existe un unique vecteur d'erreur e de poids de Hamming 1 tel que $\sigma(e) = s$

Codes de Hamming

Très peu de codes parfaits : code trivial, certains codes par répétitions, codes de Hamming, et les deux cas très spéciaux des codes de Golay

Code de Hamming = code parfait avec $d = 3$

Si C est un tel code, alors tout mot w' qui n'est pas un mot du code est à distance de Hamming 1 d'un unique mot du code
$\iff$ pour tout syndrôme non nul $s \in \mathbb{K}^{n-k}$, il existe un unique vecteur d'erreur e de poids de Hamming 1 tel que $\sigma(e) = s$

Or $\sigma(e) = e.H$ (où $H =$ est la matrice de contrôle) ;
si e est de poids 1, alors $e.H$ est simplement un multiple d'une ligne de H

Donc les lignes de H sont tous les éléments de $\mathbb{K}^{n-k} \backslash \{0\}$, à une constante multiplicative près,
i.e. (les représentants) de tous les éléments de l'espace projectif $\mathbb{P}^{n-k-1}(\mathbb{K})$

Construction de codes de Hamming binaires

1. Choisir un paramètre ℓ
2. Lister tous les mots non nuls de $\mathbb{F}_2^\ell$, en terminant par $100\ldots00,\ 010\ldots00,\ \ldots,\ 000\ldots01$
3. Faire de cette liste une matrice H de taille $(2^\ell - 1) \times \ell$ telle que $H = \begin{pmatrix} P \\ I_\ell \end{pmatrix} \in \mathcal{M}_{2^\ell-1,\ell}(\mathbb{F}_2)$
4. Générer une matrice de C : $G = \begin{pmatrix} I_{2^\ell-1-\ell} & P \end{pmatrix} \in \mathcal{M}_{2^\ell-1-\ell,2^\ell-1}(\mathbb{F}_2)$

Les paramètres du code sont $n = 2^\ell - 1$ et $k = 2^\ell - 1 - \ell$, correspondant à un code parfait avec $d = 3$.

Construction de codes de Hamming binaires

1. Choisir un paramètre ℓ
2. Lister tous les mots non nuls de $\mathbb{F}_2^\ell$, en terminant par $100\ldots00,\ 010\ldots00,\ \ldots,\ 000\ldots01$
3. Faire de cette liste une matrice H de taille $(2^\ell - 1) \times \ell$ telle que $H = \begin{pmatrix} P \\ I_\ell \end{pmatrix} \in \mathcal{M}_{2^\ell-1,\ell}(\mathbb{F}_2)$
4. Générer une matrice de C : $G = \begin{pmatrix} I_{2^\ell-1-\ell} & P \end{pmatrix} \in \mathcal{M}_{2^\ell-1-\ell,2^\ell-1}(\mathbb{F}_2)$

Les paramètres du code sont $n = 2^\ell - 1$ et $k = 2^\ell - 1 - \ell$, correspondant à un code parfait avec $d = 3$.

Exemple : $k = 3$ donne le code de Hamming classique de taille $(4, 7)$; une matrice de génératrice possible est $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

(Exercice : décoder les messages 1010101 et 0111110)

Section 4

Codes polynomiaux

Codes polynomiaux

Principe : Identifier **mots** et **polynômes** via l'isomorphisme canonique

$$\begin{array}{ccc} \mathbb{K}^n & \xrightarrow{\sim} & \mathbb{K}_{n-1}[X] \\ (m_1 m_2 \dots m_n) & \mapsto & m_1 X^{n-1} + m_2 X^{n-2} + \cdots + m_n \end{array}$$

Codes polynomiaux

Principe : Identifier **mots** et **polynômes** via l'isomorphisme canonique

$$\begin{array}{ccc} \mathbb{K}^n & \xrightarrow{\sim} & \mathbb{K}_{n-1}[X] \\ (m_1 m_2 \dots m_n) & \mapsto & m_1 X^{n-1} + m_2 X^{n-2} + \cdots + m_n \end{array}$$

Code polynomial

Un code $C \subset \mathbb{K}^n \simeq \mathbb{K}_{n-1}[X]$ est **polynomial** si il existe $g \in \mathbb{K}_{n-1}[X]$ tel que

$$C = \{P \in \mathbb{K}_{n-1}[X] \,:\, g \mid P\}$$

g est le **polynôme générateur** du code C ; ses paramètres sont n et $k = n - \deg(g)$, et

$$C = \{g\,Q \,:\, Q \in \mathbb{K}_{k-1}[X]\}$$

Codes polynomiaux : encodages

Message $m \leftrightarrow$ polynôme $Q \in \mathbb{K}_{k-1}[X]$

- Encodage naturel : $Q \mapsto g\,Q$
- Encodage systématique : $Q \mapsto X^{n-k}Q - R_Q$ avec R_Q le reste de la division euclidienne de $X^{n-k}Q$ par g.

Codes polynomiaux : encodages

Message $m \leftrightarrow$ polynôme $Q \in \mathbb{K}_{k-1}[X]$

- Encodage naturel : $Q \mapsto g\,Q$
- Encodage systématique : $Q \mapsto X^{n-k}Q - R_Q$ avec R_Q le reste de la division euclidienne de $X^{n-k}Q$ par g.

Exemple avec $g = X^3 + X + 1$ ($\leftrightarrow 1011$) et $n = 7$

Encodage naturel : matrice génératrice

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

lignes correspondent à X^3g, X^2g, Xg et g

Encodage systématique : matrice génératrice standard

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

on retrouve un code de Hamming $(4, 7)$

Syndrômes

Soit C un code polynomial de paramètre (k, n), engendré par g.
Message reçu $m \leftrightarrow$ polynôme P.

m est un mot du code $\iff g \mid P$

Deux vérifications / calculs de syndrômes possibles :

- prendre pour $\sigma(P)$ le reste de la division euclidienne de P par g
- si g est scindé à racines simples $x_1, \ldots, x_{n-k}$ prendre $\sigma(P) = (P(x_1), \ldots, P(x_{n-k}))$
  (peut être intéressant même si les x_i sont dans une extension)

Codes cycliques

Un code $C \in \mathbb{K}^n$ est **cyclique** s'il est invariant par rotation des coordonnées :

$$(m_1 \quad m_2 \quad \dots \quad m_{n-1} \quad m_n) \in C \iff (m_2 \quad m_3 \quad \dots \quad m_n \quad m_1) \in C$$

Théorème

Un code $C \in \mathbb{K}^n$ est cyclique si et seulement si il est polynomial, de générateur g qui divise $X^n - 1$.

Très utilisés $\rightsquigarrow$ CRC (cyclic redundancy check) souvent utilisé comme synonyme de "bits de contrôle"

Codes cycliques

Un code $C \in \mathbb{K}^n$ est **cyclique** s'il est invariant par rotation des coordonnées :

$$(m_1 \quad m_2 \quad \dots \quad m_{n-1} \quad m_n) \in C \iff (m_2 \quad m_3 \quad \dots \quad m_n \quad m_1) \in C$$

Théorème

Un code $C \in \mathbb{K}^n$ est cyclique si et seulement si il est polynomial, de générateur g qui divise $X^n - 1$.

Très utilisés $\rightsquigarrow$ CRC (cyclic redundancy check) souvent utilisé comme synonyme de "bits de contrôle"

Preuve : sens $\Leftarrow$ vient du fait que sur les polynômes, tourner d'un cran revient à faire $P \mapsto XP \bmod X^n - 1$

Codes cycliques

Un code $C \in \mathbb{K}^n$ est **cyclique** s'il est invariant par rotation des coordonnées :

$$(m_1 \quad m_2 \quad \dots \quad m_{n-1} \quad m_n) \in C \iff (m_2 \quad m_3 \quad \dots \quad m_n \quad m_1) \in C$$

Théorème

Un code $C \in \mathbb{K}^n$ est cyclique si et seulement si il est polynomial, de générateur g qui divise $X^n - 1$.

Très utilisés $\rightsquigarrow$ CRC (cyclic redundancy check) souvent utilisé comme synonyme de "bits de contrôle"

Preuve : $\Rightarrow$: on prend w le mot de code non nul commençant par le plus de zéros, g le polynôme (unitaire) correspondant, et $k = n - \deg(g)$. En faisant tourner w, on obtient les polynômes $g, Xg, \dots, X^{k-1}g$ qui sont tous dans C, donc C est polynomial engendré par g. On tourne encore un cran, on obtient $X^k g - X^n + 1 \in C$, qui doit être un multiple de g, d'où $g \mid X^n - 1$.

Section 5

Introduction aux codes de Reed-Solomon

La borne de Singleton

Les paramètres k, n et d (et q) doivent satisfaire d'autres contraintes, en plus de la borne de Hamming

Théorème (borne de Singleton)

Soit C un (k, n)-code linéaire. Alors $d(C) \leqslant n - k + 1$.

Preuve : quitte à permuter les coordonnées (ce qui ne modifie pas k, n ou $d(C)$), C a une matrice génératrice de la forme $G = (I_k P)$ avec $P \in \mathcal{M}_{k,n-k}(\mathbb{K})$. Les lignes de G sont des mots du code non nuls de poids de Hamming plus petit ou égal à $n - k + 1$, donc $d(C) \leqslant n - k + 1$.

La borne de Singleton

Les paramètres k, n et d (et q) doivent satisfaire d'autres contraintes, en plus de la borne de Hamming

Théorème (borne de Singleton)

Soit C un (k, n)-code linéaire. Alors $d(C) \leqslant n - k + 1$.

Preuve : quitte à permuter les coordonnées (ce qui ne modifie pas k, n ou $d(C)$), C a une matrice génératrice de la forme $G = (I_k P)$ avec $P \in \mathcal{M}_{k,n-k}(\mathbb{K})$. Les lignes de G sont des mots du code non nuls de poids de Hamming plus petit ou égal à $n - k + 1$, donc $d(C) \leqslant n - k + 1$.

Les codes tels que $d(C) = n - k + 1$ sont appelés codes MDS (**maximal distance separable**).

Exercice : montrer que les seuls codes MDS non triviaux sur $\mathbb{F}_2$ sont le codes par répétition et bits de parité.

Codes de Reed-Solomon d'un point de vue interpolation

Codes de Reed-Solomon

Soient $k, n \in \mathbb{N}$ tels que $k \leqslant n$. Soient $x_1, \ldots, x_n \in \mathbb{K}$ distincts. Le code de Reed-Solomon associé à ces paramètres est

$$C = \{(P(x_1), P(x_2), \ldots, P(x_n)) \mid P \in \mathbb{K}[X],\ \deg(P) < k\}$$

(Ceci implique $\#\mathbb{K} \geqslant n \ \rightsquigarrow \ \mathbb{K}$ doit être un corps fini de taille suffisante.)

Codes de Reed-Solomon d'un point de vue interpolation

Codes de Reed-Solomon

Soient $k, n \in \mathbb{N}$ tels que $k \leqslant n$. Soient $x_1, \ldots, x_n \in \mathbb{K}$ distincts. Le code de Reed-Solomon associé à ces paramètres est

$$C = \{(P(x_1), P(x_2), \ldots, P(x_n)) \mid P \in \mathbb{K}[X],\ \deg(P) < k\}$$

(Ceci implique $\#\mathbb{K} \geqslant n \ \rightsquigarrow\ \mathbb{K}$ doit être un corps fini de taille suffisante.)

Encodage systématique ? pour encoder $(y_1, \ldots, y_k) \in \mathbb{K}^k$

1. calculer le **polynôme d'interpolation de Lagrange** P correspondant (i.e. l'unique $P \in \mathbb{K}_{k-1}[X]$ tel que $P(x_i) = y_i$ pour tout $1 \leqslant i \leqslant k$)
2. un mot du code est $(y_1, \ldots, y_k, P(x_{k+1}), \ldots, P(x_n))$.

Paramètres des codes de Reed-Solomon

Théorème

Soit $C = \{(P(x_1), P(x_2), \ldots, P(x_n)) \mid P \in \mathbb{K}_{k-1}[X]\}$ un code de Reed-Solomon.
Alors C est (k, n)-code MDS, i.e. $d(C) = n - k + 1$.

Preuve : C est l'image de $\mathbb{K}_{k-1}[X]$ par l'application $P \mapsto (P(x_1) \ldots, P(x_n))$, donc $\dim(C) \leqslant k$, et l'existence d'un encodage systématique $\mathbb{K}^k \to C$ montre que $\dim(C) \geqslant k$.

Pour tout mot du code $w = (P(x_1), \ldots, P(x_n))$, si $h(w) \leqslant n - k$ alors $P(x_i) = 0$ pour au moins k indices $i \in [\![1, n]\!]$, donc P a au moins k racines, donc $P = 0$ (et $w = 0$).
Ainsi des mots du code non nuls ont un poids de Hamming au moins égal à $n - k + 1$, donc $d(C) \geqslant n - k + 1$.
Mais la borne de Singleton donne $d(C) \leqslant n - k + 1$, d'où l'égalité.

Un décodeur pour Reed-Solomon

Capacité de correction $t = \lfloor \frac{n-k}{2} \rfloor$

Transmis : $c = (P(x_1), \ldots, P(x_n))$ avec $\deg P \leqslant k-1$, reçu : $c' = (y_1, \ldots, y_n)$.

Un décodeur pour Reed-Solomon

Capacité de correction $t = \lfloor \frac{n-k}{2} \rfloor$

Transmis : $c = (P(x_1), \ldots, P(x_n))$ avec $\deg P \leqslant k-1$, reçu : $c' = (y_1, \ldots, y_n)$.

1. Trouver deux polynômes Q_1, Q_0 tels que : $\deg Q_1 < n - t - k + 1$, $\deg(Q_0) < n - t$, et $\forall\ 1 \leqslant i \leqslant n,\ Q_0(x_i) - y_i Q_1(x_i) = 0$.

Un décodeur pour Reed-Solomon

Capacité de correction $t = \lfloor \frac{n-k}{2} \rfloor$

Transmis : $c = (P(x_1), \ldots, P(x_n))$ avec $\deg P \leqslant k-1$, reçu : $c' = (y_1, \ldots, y_n)$.

1. Trouver deux polynômes Q_1, Q_0 tels que : $\deg Q_1 < n - t - k + 1$, $\deg(Q_0) < n - t$, et $\forall\ 1 \leqslant i \leqslant n,\ Q_0(x_i) - y_i Q_1(x_i) = 0$.

   Système **linéaire** de n équations à $(n-t) + (n-t-k+1) \geqslant n+1$ inconnues (coefficients des polynômes)
   $\Rightarrow$ il existe au moins une solution (Q_0, Q_1) non triviale, donc avec $Q_1 \neq 0$ (sinon Q_0 à trop de racines donc $Q_0 = 0$)

Un décodeur pour Reed-Solomon

Capacité de correction $t = \lfloor \frac{n-k}{2} \rfloor$

Transmis : $c = (P(x_1), \ldots, P(x_n))$ avec $\deg P \leqslant k-1$, reçu : $c' = (y_1, \ldots, y_n)$.

1. Trouver deux polynômes Q_1, Q_0 tels que : $\deg Q_1 < n - t - k + 1$, $\deg(Q_0) < n - t$, et $\forall\ 1 \leqslant i \leqslant n,\ Q_0(x_i) - y_i Q_1(x_i) = 0$.

   Système **linéaire** de n équations à $(n-t) + (n-t-k+1) \geqslant n+1$ inconnues (coefficients des polynômes)
   $\Rightarrow$ il existe au moins une solution (Q_0, Q_1) non triviale, donc avec $Q_1 \neq 0$ (sinon Q_0 à trop de racines donc $Q_0 = 0$)
2. Si pas d'erreur en position i, alors $y_i = P(x_i)$ donc $Q_0(x_i) - P(x_i)Q_1(x_i) = 0$: x_i est racine de $Q_0 - PQ_1$
3. Donc si au plus t erreurs, $Q_0 - PQ_1$ a au moins $n-t$ racines, or $\deg(Q_0 - PQ_1) < n - t$, donc $Q_0 - PQ_1 = 0$ et $P = Q_0/Q_1$.

Complexité ?

Codes de Reed-Solomon comme codes polynomiaux

On choisit $(x_1, x_2, \ldots, x_n) = (1, \alpha, \ldots, \alpha^{n-1})$ avec $\alpha \in \mathbb{K}^*$ d'ordre n impose $n \mid q-1$ si $\mathbb{K} = \mathbb{F}_q$
C = code de Reed-Solomon de paramètre (k, n) associé.

Proposition

C est un code polynomial, de polynôme générateur $g = \prod_{m=k}^{n-1}(X - \alpha^m)$

Preuve : Soit $w = (P(1), \ldots, P(\alpha^{n-1}))$ un mot du code, avec $P = \sum_{0 \leqslant j < k} c_j X^j$. Le polynôme associé à w est $Q = \sum_{0 \leqslant i < n} P(\alpha^i) X^{n-1-i}$. Alors pour tout $k \leqslant m \leqslant n-1$, $Q(\alpha^m) = \sum_{0 \leqslant i < n} (\sum_{0 \leqslant j < k} c_j (\alpha^i)^j)(\alpha^m)^{n-1-i} =$
$\sum_{0 \leqslant j < k} c_j \alpha^{m(n-1)} \sum_{0 \leqslant i < n} (\alpha^{j-m})^i) = \sum_{0 \leqslant j < k} c_j \alpha^{m(n-1)} \frac{(\alpha^{j-m})^n - 1}{\alpha^{j-m} - 1} = 0$.
Donc $g \mid Q$, d'où C inclus dans le code engendré par g ; on conclut par égalité des dimensions.

Point de vue polynomial : un autre décodeur

Soit $Q \in \mathbb{K}_{n-1}[X]$ le polynôme correspondant au message transmis, $\tilde{Q} = Q + E$ celui correspondant au message reçu.

Polynôme d'erreur $E = \sum_{i \in I} c_i X^i$ avec I ensemble des positions des erreurs.

Syndrôme : $\sigma(\tilde{Q}) = \sigma(E) = (\tilde{Q}(\alpha^k), \ldots, \tilde{Q}(\alpha^{n-1})) \in \mathbb{K}^{n-k}$

Point de vue polynomial : un autre décodeur

Soit $Q \in \mathbb{K}_{n-1}[X]$ le polynôme correspondant au message transmis, $\tilde{Q} = Q + E$ celui correspondant au message reçu.

Polynôme d'erreur $E = \sum_{i \in I} c_i X^i$ avec I ensemble des positions des erreurs.

Syndrôme : $\sigma(\tilde{Q}) = \sigma(E) = (\tilde{Q}(\alpha^k), \ldots, \tilde{Q}(\alpha^{n-1})) \in \mathbb{K}^{n-k}$

Propriété

La séquence $(\tilde{Q}(\alpha^k), \ldots, \tilde{Q}(\alpha^{n-1}))$ vérifie une relation de récurrence linéaire d'ordre $|I|$, de polynôme caractéristique $\prod_{i \in I}(X - \alpha^i)$

Ce polynôme s'appelle le **polynôme localisateur**

- si $|I| \leqslant t = \lfloor (n-k)/2 \rfloor$, on peut le retrouver à partir des $n-k$ valeurs $\tilde{Q}(\alpha^k), \ldots, Q(\alpha^{n-1})$
- une fois connu on peut facilement trouver I puis corriger les erreurs

Remarques

- Les codes de Reed-Solomon sont très courants : utilisés dans les CD, DVD, QR codes, ADSL,...
- Avec le choix de $x_1, \ldots, x_n$ le rendant polynomial, on obtient un cas particulier de codes BCH (Bose–Ray-Chaudhuri–Hocquenghem), dont la procédure de décodage est celle que l'on vient d'esquisser
- Trouver le polynôme minimal d'une suite récurrente linéaire d'ordre n peut se faire en $O(n^2)$ avec l'algorithme de Berlekamp-Massey (utile aussi pour cryptanalyse LFSR !)