

Cours 6 Codes parfaits

- Inégalité de Hamming [Richard Hamming (1915–1998), mathématicien américain]

On se donne un code linéaire (ou éventuellement non linéaire)  $\varphi$  de dimension  $k$  et de longueur  $n \geq k$ . On a  $u = \varphi(a)$  avec  $a \in (\mathbb{F}_2)^k$  et  $u \in (\mathbb{F}_2)^n$ . L'ensemble des mots du code est noté  $\mathcal{C}$ . Avec les notations précédentes, si le code  $\varphi$  corrige  $t$  erreurs sans ambiguïté, alors on a  $\sum_{j=0}^{j=t} \binom{n}{j} \leq 2^{n-k}$ .

Cette inégalité s'appelle aussi "inégalité de Hamming". Commençons par la vérifier pour les exemples déjà vus.

Exemple 1. Duplication d'un bit :  $k = 1, n = 2, t = 0$ :  $\sum_{j=0}^{j=0} \binom{2}{j} = 1 \leq 2^{2-1} = 2$ .

Exemple 2. Double répétition d'un bit :  $k = 1, n = 3, t = 1$ :  $\sum_{j=0}^{j=1} \binom{3}{j} = 1 + 3 = 4 \leq 2^{3-1} = 4$ . On constate qu'on est dans le cas où l'inégalité de Hamming est en fait une égalité.

Exemple 3. Duplication de deux bits :  $k = 2, n = 4, t = 0$ :  $\sum_{j=0}^{j=0} \binom{4}{j} = 1 \leq 2^{4-2} = 4$ .

Exemple 4. Contrôle de parité :  $k = 3, n = 4, t = 0$ :  $\sum_{j=0}^{j=0} \binom{4}{j} = 1 \leq 2^{4-3} = 2$ .

Exemple 5. Code de Hamming H7 :  $k = 4, n = 7, t = 1$ :  $\sum_{j=0}^{j=1} \binom{7}{j} = 1 + 7 = 8 \leq 2^{7-4} = 8$ . Ici encore, l'inégalité de Hamming est une égalité.

- Preuve de l'inégalité de Hamming

Si  $u \in \mathcal{C}$  est un mot du code, on note  $B(u, t)$  la "boule fermée" de centre  $u$  et de rayon  $t$  pour la distance de Hamming :  $B(u, t) = \{v \in (\mathbb{F}_2)^n, d(u, v) \leq t\}$ .

Si le code  $\mathcal{C}$  corrige  $t$  erreurs sans ambiguïté, les boules  $B(u, t)$  et  $B(\tilde{u}, t)$  sont d'intersection vide si  $u$  et  $\tilde{u}$  sont deux mots différents du code. En effet, si le code corrige  $t$  erreurs, on a vu que sa distance minimale  $d$  est supérieure ou égale à  $2t + 1$ . Donc si  $v \in B(u, t)$  et  $u \neq \tilde{u}$  appartiennent à  $\mathcal{C}$ , on a  $2t + 1 \leq d \leq d(u, \tilde{u}) \leq d(u, v) + d(v, \tilde{u}) \leq t + d(v, \tilde{u})$  et le point  $v \in B(u, t)$  ne peut pas appartenir à la boule fermée  $B(\tilde{u}, t)$  de centre  $\tilde{u}$  et de rayon  $t$ .

On compte ensuite le nombre de points de l'espace discret  $(\mathbb{F}_2)^n$  à l'intérieur de la boule  $B(u, t)$ . Il y a d'abord  $u$  lui-même et les points  $v \in \mathcal{C}$  qui diffèrent de  $u$  de un bit exactement ; il y en a au total  $\binom{n}{1} = n$ . Il y a ensuite les points qui diffèrent de  $u$  de deux bits exactement ; on en compte  $\binom{n}{2} = \frac{n(n-1)}{2}$ . On dispose de  $\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$  points  $v \in \mathcal{C}$  qui diffèrent de  $u$  de trois bits exactement. Et ainsi de suite. On a en général  $\binom{n}{t} = \frac{n(n-1)\dots(n-(t-1))}{t!}$  points qui diffèrent de  $u$  de  $t$  bits exactement. Donc le nombre de points  $|B(u, t)|$  dans la boule  $B(u, t)$  est égal à la somme des nombres précédents et on a  $|B(u, t)| = \sum_{j=0}^{j=t} \binom{n}{j}$ .

Comme les boules  $B(u, t)$  sont toutes disjointes pour tout  $u \in \mathcal{C}$ , le nombre total de points  $v \in (\mathbb{F}_2)^n$  qui appartiennent au moins à une telle boule est égal à  $\sum_{u \in \mathcal{C}} |B(u, t)| = 2^k \sum_{j=0}^{j=t} \binom{n}{j}$  car il y a exactement  $2^k$  points dans l'ensemble des mots d'un code de dimension  $k$ .

Enfin, la réunion de toutes les boules  $B(u, t)$  pour tous les  $u \in \mathcal{C}$  est incluse dans l'espace discret  $(\mathbb{F}_2)^n$ , qui compte lui  $2^n$  points. On en déduit  $2^k \sum_{j=0}^{j=t} \binom{n}{j} \leq 2^n$ , qui montre l'inégalité proposée après division par  $2^k$ .

- Code parfait

Un code  $\varphi: (\mathbb{F}_2)^k \rightarrow (\mathbb{F}_2)^n$  qui corrige sans ambiguïté  $t$  erreurs est parfait si et seulement si l'inégalité de Hamming est une égalité :  $\sum_{j=0}^{j=t} \binom{n}{j} = 2^{n-k}$ .

Pour un code parfait, tout mot reçu  $v \in (\mathbb{F}_2)^n$  appartient à au moins une boule  $B(u, t)$  centrée en  $u \in \mathcal{C}$ . Comme ces boules sont disjointes, cette boule  $B(u, t)$  est unique et il existe un unique mot du code  $u$  de sorte que sa distance à tout mot  $v$  reçu soit inférieure ou égale à  $t$  :

$\forall v \in (\mathbb{F}_2)^n, \exists ! u \in \mathcal{C}, d(u, v) \leq t$ . La correction d'un message reçu  $v$  contenant au plus  $t$  erreurs s'effectue alors toujours sans ambiguïté. On parle parfois de projection du mot reçu sur le code  $\mathcal{C}$  et on note  $u = \Delta(v)$  l'unique mot du code à distance minimale du mot  $v$  reçu.

- Cas du code de double répétition d'un bit

La liste des mots du code permet une détermination quasi-immédiate de la projection  $\Delta(v)$ .

valeur du mot reçu $v$	commentaire
000	distance $\leq 1$ du mot 000
001	
010	
100	
011	distance $\leq 1$ du mot 111
101	
110	
111	

Pour le code de Hamming H7, la liste des 16 mots du code et celle des 128 possibilités de réception d'un mot de 7 bits est plus longue à écrire.

- Codes parfaits qui corrigent une erreur

Dans le cas où  $t = 1$ , on a l'égalité  $1 + n = 2^{n-k}$ . Pour les différentes valeurs de  $r \equiv n - k$ , on détermine successivement  $n = 2^r - 1$  puis  $k = n - r$ . Si  $1 \leq r \leq 6$ , on le tableau suivant

$r$	1	2	3	4	5	6
$n$	1	3	7	15	31	63
$k$	0	1	4	11	26	57
rendement $\frac{k}{n}$	0	$\frac{1}{3}$	$\frac{4}{7} \simeq 0,571$	0,733	0,839	0,905

Pour  $r = 2$ , on retrouve la double duplication de 1 bit et pour  $r = 3$ , le code de Hamming H7.

- Unicité

Si un code parfait linéaire corrige une erreur exactement, c'est à dire si  $n = 2^r - 1$  avec  $r$  entier supérieur ou égal à 1 et  $k = n - r$ , alors il est unique à une permutation près des colonnes de sa matrice de contrôle  $H$ .

- Code de Hamming  $H_n$

Si  $n$  est un entier de la forme  $n = 2^r - 1$  et  $r$  un entier supérieur ou égal à 1, le code de Hamming  $H_n$  est par définition le code parfait linéaire de longueur  $n$  et de dimension  $k = n - r$  qui corrige exactement une erreur.

On appelle efficacité d'un code correcteur la probabilité d'erreur résiduelle après correction. On note ici que l'efficacité des codes de Hamming décroît avec l'entier  $n$ .

On se place dans le cas d'un canal symétrique et sans mémoire tel que la probabilité de transition  $0 \rightarrow 1$  ou  $1 \rightarrow 0$  est égale à  $p$ , avec  $0 < p < 1$ . On note  $P_\ell$  la probabilité d'avoir  $\ell$  erreurs lors de la transition d'un mot de  $n$  bits. On a alors  $P_\ell = \binom{n}{\ell} p^\ell (1-p)^{n-\ell}$ . Pour le code de Hamming  $H_n$ , l'efficacité vaut  $\varepsilon = 1 - P_0 - P_1$ . On la calcule pour les codes parfaits introduits plus haut, dans le cas  $p = 10^{-2}$ :

$r$	2	3	4	5	6
$n$	3	7	15	31	63
$P_0$	0,97	0,93	0,86	0,73	0,53
$P_1$	0,029	0,066	0,13	0,23	0,34
$\varepsilon$	$3,0 \cdot 10^{-4}$	$2,0 \cdot 10^{-3}$	$9,6 \cdot 10^{-3} \simeq 1\%$	$0,0038 \simeq 4 \%$	$0,13 \simeq 13 \%$

Notons par exemple que le code  $H_{63}$  envoie des paquets de 63 bits dont 57 utiles avec seulement 6 bits de contrôle. On constate que même après correction, on se trompe dans l'interprétation du message reçu dans 13% des cas !

- Recherche d'un code parfait linéaire qui corrige deux erreurs

Pour  $t = 2$ , la relation d'égalité pour l'inégalité de Hamming s'écrit  $1 + n + \frac{n(n-1)}{2} = 2^r$ , avec  $r$  entier  $\geq 1$ . Il s'agit d'une équation du second degré d'inconnue  $n$  qui doit avoir une solution entière ; l'entier  $n$  doit être un nombre entier.

Pour  $n = 5$ , on a  $1 + 5 + 10 = 16 = 2^4$ . On a 4 bits de contrôle pour un message de dimension  $k = n - 4 = 1$  ! Le codage s'écrit très simplement :  $0 \mapsto 00000$  et  $1 \mapsto 11111$ . On corrige bien deux bits puisque  $d = 5 \geq 2 \times 2 + 1$ .

Il n'y a pas d'autre solution pour  $n \leq 20$ .

- codes parfaits linéaires qui corrigent trois erreurs

Le cas d'égalité pour  $t = 3$  s'écrit  $1 + n + \frac{1}{6}n(n-1) + \frac{1}{6}n(n-1)(n-2) = 2^r$ , avec  $r$  entier  $\geq 1$ . On a cette fois une équation du troisième degré d'inconnue  $n$  qui doit avoir une solution entière. Le plus simple est de regarder si l'expression  $\sum_{j=0}^{t-1} \binom{n}{j}$  est une puissance de 2 lorsque l'entier  $n$  est fixé.

Pour  $n = 7$ , on a  $1 + 7 + 21 + 35 = 64 = 2^6$ . Donc  $k = 7 - 6 = 1$  et on duplique un unique bit six fois de suite :  $0 \mapsto 0000000$  et  $1 \mapsto 1111111$ . On a bien  $d = 7 \geq 2 \times 3 + 1$  mais le rendement de  $\frac{1}{7} \simeq 0,14$  est très faible.

- Code de Golay (1949)

[Marcel Golay (1902–1989), mathématicien suisse]

Avec Golay, on remarque que

$\sum_{j=0}^{j=3} \binom{23}{j} = 1 + 23 + (23 \times 11) + (23 \times 11 \times 7) = 1 + 23 + 253 + 1771 = 2048 = 2^{11}$ . Donc  $k = 23 - 11 = 12$ . Pour envoyer 12 bits "utiles", on en transmet en fait 23. Le rendement est modeste :  $\frac{k}{n} = \frac{12}{23} \simeq 0,52$  mais loin d'être ridicule ! La matrice de parité  $P$  contient 12 lignes

et 11 colonnes. La matrice de contrôle s'écrit  $H = (P^t \ I_{11})$  et la transposée  $P^t$  de la matrice de parité est une matrice de 11 lignes et 12 colonnes qui satisfait à la relation suivante :

$$P^t = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

L'efficacité  $\varepsilon = 1 - P_0 - P_1 - P_2 - P_3$  est très bonne, même pour  $p = \frac{1}{100}$ . On a en effet le tableau suivant :

$P_\ell$	valeur exacte	valeur numérique approchée pour $p = \frac{1}{100}$
$P_0$	$(1 - p)^{23}$	0,7936
$P_1$	$23 p (1 - p)^{22}$	0,1844
$P_2$	$253 p^2 (1 - p)^{21}$	0,0205
$P_3$	$1771 p^3 (1 - p)^{20}$	$1,45 \cdot 10^{-3}$

D'où  $\varepsilon \simeq 7,610^{-5}$ .

## Exercices

- Codage par blocs

On code un mot  $abc$  de trois bits de la façon suivante :  $u = abccbadef$ , avec  $d = b + c$ ,  $e = a + c$  et  $f = b + a$ .

- Donner la dimension  $k$  et la longueur  $n$  de ce code. Quel est son rendement ?
- Montrer que ce code est linéaire.
- Écrire la liste des mots du code.
- Quelle est la distance minimale de ce code ? [4]
- Combien d'erreurs peut-il détecter de façon certaine ? [ $d - 1 = 3$ ]
- Combien d'erreurs peut-il éventuellement corriger sans ambiguïté ? [1]
- Ce code est-il parfait ?
- Est-ce un code de Hamming ?

- Codes de longueur 15

On considère un code  $C(15,8)$  de longueur  $n = 15$  et de dimension  $k = 8$ . On transmet les mots codés au moyen d'un canal symétrique (sans mémoire) et la probabilité pour qu'un bit soit mal transmis vaut  $p = 0,02$ .

## CODES ET AUTOMATES FINIS

- a) On transmet un mot de code. Quelle est la probabilité pour que :
- (i) le mot soit bien transmis ?  $[P_0 = (1 - p)^{15} \simeq 0,738569]$
  - (ii) seul le premier bit soit mal transmis ?
  - (iii) le mot soit transmis avec exactement une erreur ?  
 $[P_1 = 15 p (1 - p)^{14} \simeq 0,226093]$
  - (iv) le mot soit transmis avec exactement deux erreurs ?  
 $[P_2 = 105 p^2 (1 - p)^{13} \simeq 0,032299]$
  - (v) le mot soit transmis avec au moins trois erreurs ?  
 $[1 - P_0 - P_1 - P_2 \simeq 0,003039]$
- b) Vérifier qu'un code  $C(15, 8)$  peut corriger une erreur. Est-ce alors un code parfait ?
- c) On suppose que ce code corrige une erreur. Quelle est la probabilité pour qu'un message soit mal corrigé ? [environ 3,5 %]
- d) Déterminer la valeur de  $k$  pour qu'un code  $C(15, k)$  puisse être parfait en corrigeant une erreur.  $[k = 11]$
- e) Vérifier qu'un code  $C(15, 8)$  peut corriger deux erreurs. Est-ce alors un code parfait?  $[121 < 128]$
- f) On suppose que ce code corrige deux erreurs. Quelle est la probabilité pour qu'un message soit bien corrigé ?  $[P_0 + P_1 + P_2 \simeq 0,996961]$

On rappelle l'inégalité de Hamming. Si un code  $C(n, k)$  corrige  $t$  erreurs, alors

$$\sum_{p=0}^t \binom{n}{p} \leq 2^r, \text{ avec } n = k + r.$$