Arithmétique et applications, graphes et combinatoire

Cours No. 2, Codes correcteurs 1 : codes linéaires et codes de Hamming

Quand on transmet de l'information, le message reçu n'est pas toujours le même que le message envoyé. Un code correcteur est un code pour lequel on peut detecter et corriger des erreurs qui surviennent durant la transmission. Il s'agit d'une technique de codage basée sur la redondance. Ces codes sont appliqués pour la transmission d'information par l'internet, des données stockées sur un ordinateur ou sur un disque compact, transmission d'images d'un vaisseau spacial...

Par exemple, supposons qu'on encrypte de l'information par un alphabet binaire $\{0,1\}$, mais au lieu d'utiliser un seul bit (élément binaire) d'information, on utilise trois bits:

$$0 \mapsto 000, \qquad 1 \mapsto 111\,.$$

Si l'effet de bruit est d'inverser un bit $0 \leftrightarrow 1$ avec probabilité p, alors un seul interchange parmie les trois bits se passe avec probabilité $3p(1-p)^2$, deux interchanges avec probabilité $3p^2(1-p)$ et trois avec probabilité p^3. Il s'ensuit que la règle de la majorité donne le bon résultat avec probabilité

$$1 - p^3 - 3p^2(1-p) = (1-p)^2(1+2p)\,.$$

Si $p = 10^{-2}$ la probabilité d'erreur est alors $1-(1-p)^2(1+2p) = p^2(3-2p) \sim 3\times 10^{-4}$. Pourtant, avec un seul bit la probabilité d'erreur est 10^{-2}.

Soit K un corps. Alors $K^n = \{(a_1, a_2, \ldots, a_n) | a_j \in K\}$ est un espace vectoriel sur K. Par exemple, ${\mathbb{F}_3}^2 = (\mathbf{Z}/3\mathbf{Z})^2$ est un espace vectoriel à deux dimensions dont les composantes de chaque vecteur appartient à $\mathbf{Z}/3\mathbf{Z} = \{0,1,2\}$. Il existe $3\times 3 = 9$ tels vecteurs :

$$(0,0),\ (0,1),\ (0.2),\ (1,0),\ (1,1),\ (1,2),\ (2,0),\ (2,1),\ (2,2),$$

On prend la somme et on multiplie par un scaleur en appliquant l'addition et multiplication modulo 3 :

$$(1,2) + (1,1) = (2,0), \qquad 2\cdot(2,1) = (1,2)\,.$$

Terminologie :

Alphabet : un ensemble fini non vide dont ses éléments sont appelés *lettres* ou *symboles* ; un alphabet binaire contient deux symboles $\{0,1\} = \mathbb{F}_2$.

Message ou mot : une suite à valeurs dans un alphabet.

Code : Une collection de mots admissibles.

Longueur d'un message : le nombre de lettres que le message contient.
Code en bloc : un code correcteur traitant des messages de longueur fixe – pour transmettre un message quelconque une solution consiste à concaténer une suite de blocs.

On va considérer des codes pour lesquels les mots sont de longueur fixe appartenant à $\mathbb{F}_2{}^n = (\mathbf{Z}/2\mathbf{Z})^n$ pour un n donné.
Un code C est un sous-ensemble de $\mathbb{F}_2{}^n$. Si C est un sous-espace linéaire on dit que C est un code *linéaire*.

Correction d'erreurs au plus proche voisin (ou correction par proximité) : On adopte la stratégie de corriger au plus proche voisin dans C. Par exemple, soit $C = \{(1010), (1110), (0011)\} \subset \mathbb{F}_2{}^4$. Supposons qu'un mot de C est transmis et on reçoit le vecteur $r = (0110)$. On voit que le mot $c = (1110)$ est celui que diffère de r dans le moins de positions et par suite on corrige r en c en supposant que l'erreur en r est $e = r - c = (1000)$.

Par contre, si on reçoit $s = (0010)$, on voit qu'il y a deux mots de C qui diffère en une seule position, donc il n'existe pas de voisin unique et le mot n'est pas corrigible. On peut règler ce problème avec un choix judicieux de C :

Distance de Hamming : Soit $C \subset \mathbb{F}_2{}^n$. Pour $x, y \in C$ on définit la distance de Hamming entre x et y comme le nombre de positions dont les deux mots diffèrent : $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$. On écrit $d = d(C)$ pour la distance minimale entre deux mots de C. Dans l'exemple ci-dessus on a $d = 1$.

Pour $x \in \mathbb{F}_2{}^n$ et pour un entier positif r, soit $S_r(x) = \{y \in \mathbb{F}_2{}^n | d(x, y) < r\}$ la *boule de rayon r centré en x*. Soit C un code avec distance minimale d et soit t le plus grand entier tel que $t < d/2$. Il s'ensuit que $S_t(x) \cap S_t(y)$ est vide pour chaque paire de mots distincts $x, y \in C$. Donc si le vecteur $z \in \mathbb{F}_2{}^n$ est reçu et $d(u, z) \leq t$ pour un $u \in C$, il s'ensuit que $z \neq S_t(v)$ pour tout autre $v \in C$. Autrement dit, si un vecteur z diffère d'un mot $u \in C$ en $\leq t$ positions, alors chaque autre mot dans C nécessairement diffère de z en plus de t positions. Donc la stratégie du plus proche voisin permettra la correction de t ou moins d'erreurs. On dit que le code C est *t–correcteur*.

Exemple : Soit $C = \{(00000000), (11100011), (00011111), (11111100)\}$. Alors la distance minimale de C est $d = 5$. Puisque $t = 2$ est le plus grand entier tel que $t < d/2$, alors C est 2-correcteur.

Soit C un code t-correcteur dans $\mathbb{F}_2{}^n$. On voudrait connaître le nombre de vecteurs dans $\mathbb{F}_2{}^n$ qui sont corrigible en C. Observons d'abord que pour tout $x \in \mathbb{F}_2{}^n$, il existe

$\begin{pmatrix} n \\ i \end{pmatrix}$ vecteurs dans $\mathbb{F}_2{}^n$ qui diffère de x en exactement i positions. De plus, tout vecteur dans $\mathbb{F}_2{}^n$ qui diffère de x en i positions appartient à $S_t(x)$ si et seulement si $i \leq t$. Il s'snsuit que le nombre de vecteurs dans $S_t(x)$ est

$$\begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} n \\ t \end{pmatrix}.$$

Afin de déterminer le nombre de vecteurs dans $\mathbb{F}_2{}^n$ qui sont corrigibles, il suffit de compter le nombre de vecteurs dans $S_t(x)$ lorsque x prend ses valeurs dans C. Puisque les ensembles $S_t(x)$ sont disjoints, on obtient

$$|C| \left\{ \begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} n \\ t \end{pmatrix} \right\}$$

Puisque $|\mathbb{F}_2{}^n| = 2^n$ on obtient la *borne de Hamming* :

Théorème 1 : Soit C un code t-correcteur dans $\mathbb{F}_2{}^n$. Alors

$$|C| \left\{ \begin{pmatrix} n \\ 0 \end{pmatrix} + \begin{pmatrix} n \\ 1 \end{pmatrix} + \cdots + \begin{pmatrix} n \\ t \end{pmatrix} \right\} \leq 2^n .$$

Un code s'appelle *parfait* si chaque vecteur dans $\mathbb{F}_2{}^n$ est corrigible en C, c'est à dire on a égalité dans l'inégalité ci-dessus. Dans l'exemple on a $|C| = 4$,

$$\begin{pmatrix} 8 \\ 0 \end{pmatrix} + \begin{pmatrix} 8 \\ 1 \end{pmatrix} + \begin{pmatrix} 8 \\ 2 \end{pmatrix} = 37$$

et $2^8 = 256$. Ce code est loin d'être parfait ! Dans la suite on rencontrera des codes parfaits, notamment les codes de Hamming.

Parfois il est utile de former la matrice dont les lignes sont les éléments de C. Pour l'example ci-dessus, il s'agit de

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

Théorème 2 : Soit r le nombre de mots dans un code avec paramètres (n, d) tel que $d > n/2$. Alors $r \leq \frac{2d}{2d-n}$.

Exercice : Pour le code

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

calculer n, d et t. Est-ce-que ce code est parfait ? Est-ce-qu'il est linéaire ?

Codes de Hadamard : Une $n \times n$ matrice R s'appelle une *matrice de Hadamard* si les coefficients sont tous 1 où -1 et $RR^t = nI$ (I la matrice identité). Il s'ensuit que $R^{-1} = \frac{1}{n}R^t$ et que $R^tR = nI$, d'où le produit scalaire de chaque ligne ou colonne avec lui-même égale n et le produit scalaire de deux lignes distinctes ou deux colonnes distinctes égale 0. Par conséquence, le changement de signe d'une ligne où d'une colonne donne une autre matrice de Hadamard. On peut s'arranger alors que tous les coefficients de la première ligne et de la première colonne sont 1. Dans ce cas on dit que la matrice de Hadamard est *normalisée*. Puisque la première ligne et colonne d'une matrice de Hadamard normalisée contient que le coefficient 1, on en déduit que les autres lignes et colonnes contiennent le même nombre de 1 et de -1, par suite n est paire. En fait n est un multiple de 4, car si $R = (r_{ij})$,

$$\sum_j (r_{1j} + r_{2j})(r_{1j} + r_{3j}) = \sum_j {r_{1j}}^2 = n$$

et $(r_{1j} + r_{2j}(r_{1j} + r_{3j}) = 0$ ou 4 pour chaque j.

On peut obtenir des $2^n \times 2^n$ matrices de Hadamard normalisée comme suite :

$$R_1 = (1) \quad R_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad R_4 = \begin{pmatrix} R_2 & R_2 \\ R_2 & -R_2 \end{pmatrix} \quad \cdots R_{2^n} = \begin{pmatrix} R_{2^{n-1}} & R_{2^{n-1}} \\ R_{2^{n-1}} & -R_{2^{n-1}} \end{pmatrix}$$

Afin de convertir les matrices de Hadamard en un code on enlève la première ligne et colonne et on remplace -1 par 0. Par exemple, à partir de R_8 on obtient

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

En général, si le rang d'une matrice de Hadamard est $4m \geq 8$, les $4m-1$ lignes de la matrice A sont des mots de longueur $4m-1$ contenant $2m$ zéros et $2m-1$ uns. La distance minimale entre mots est $d = 2((2m-1)-(m-1)) = 2m$ (exercice). En ajoutant le vecteur $(11\cdots1)$ de longueur $4m-1$ on obtient un code C avec $4m$ mots de longueur $4m-1$ avec distance minimale $2m$. Une conséquence du Théorème 2 est le fait qu'il est impossible d'ajouter des mots à ce code sans diminuer la distance minimale d. On appelle ces codes des *codes de Hadamard*.

Codes de Reed-Muller : Soit A la matrice construite ci-dessus à partir d'une matrice de Hadamard normalisée de rang $4m$ et soit B la matrice qui en résulte en interchangeant tous les zéros et uns dans A. Soit $\mathcal{A}$ la matrice obtenue de A en plaçant un 1 au début de chaque ligne de A et soit $\mathcal{B}$ la matrice obtenue de B en plaçant un 0 devant chaque ligne de B. Alors l'ensemble des lignes de $\mathcal{A}$ et $\mathcal{B}$ donne un code contenant $8m-2$ mots de longueur $4m$ avec distance minimal $2m$ (exercice). Il s'appelle un code de Reed-Muller. Un tel code contenant 64 mots était utilisé dans le vaisseau spacial Mariner 9 en 1972 qui prenait des images de la planète Mars. Avant transmission, chaque photographe était décomposée en une collection de petits points, chaqu'un accordé un niveau de gris en correspondance avec les 64 mots.

Codes linéaires : Un problème évident est de trouver le mot le plus proche d'un mot reçu contenant des erreurs. Pour l'instant notre seul méthode est de parcourir les différents mots dans C, ce qui est difficile lorsque $|C|$ est grand. Pour les codes linéaires on peut faire mieux.

On suppose $C \subset (\mathbf{Z}/2\mathbf{Z})^n$ est un sous-espace vectoriel de dimension k (voir l'exercice ci-dessus). Soit $W = (\mathbf{Z}/2\mathbf{Z})^k$ et soit $V = (\mathbf{Z}/2\mathbf{Z})^n$ avec $k < n$ et soit G une $(k \times n)$–matrice sur $\mathbf{Z}/2\mathbf{Z}$ de rang k (les k lignes sont indépendantes). Alors

$C := \{v \in V \,|\, \exists w \in W \text{ t.q. } v = wG\}$ est un sous-espace de V de dimension k. Le code C contient 2^k mots (le cardinal de W). La matrice G s'appelle la *matrice génératrice* de C.

Exemple : Soit $W = (\mathbf{Z}/2\mathbf{Z})^2 = \{(0,0),(1,0),(0,1),(1,1)\}$ et soit

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

Alors $C = \{(00000000000),(11110000111),(00001111111),(11111111000)\}$ est le code qui en résulte avec $n = 11$ et $k = 2$. Pour ce code $d = 7$ et donc il est 3-correcteur.

La pratique : On encrypte un message avec W, puis on convertit en des mots de C en appliquant G. On transmet le message en corrigant des erreurs si besoin, enfin on reconvertit en des mots de W. Puisque le rang de G est maximal, il existe une (11×2)–inverse à droite B : $GB = I_2$ de telle sorte que $w = wGB$, ce qui nous permet de récuperer le message.

Détéction d'erreurs dans un code linéaire : On suppose C un code construit à partir d'une $(k \times n)$–matrice génératrice G. Supposons qu'on peut trouver une $(n-k) \times n$ – matrice H de rang maximal $n-k$ telle que $HG^t = 0$. Dans ce cas pour tout $w \in W$:

$$HG^t w^t = 0 \;\Rightarrow\; H(wG)^t = 0 \;\Rightarrow\; Hc^t = 0 \;\forall c \in C\,.$$

Puisque le rang de H est maximal, une comparaison de dimensions montre que $Hc^t = 0 \;\Leftrightarrow\; c \in C$. On peut alors appliquer H afin d'identifier les mots de C. On appele H la *matrice de contrôle* ou la *matrice de parité*.

Afin de détérminer la matrice de contrôle H, on voit que $HG^t = 0 \;\Rightarrow\; GH^t = 0$ d'où les colonnes de H^t (les lignes de H) sont dans le noyau de G. Il suffit alors de trouver une base pour le noyau de G de telle sorte que les lignes de H sont les éléments de cette base. Dans la pratique, on peut commencer avec H et puis détérminer G à partir de $HG^t = 0$ de la même manière : On trouve une base pour le noyau de H et on met les éléments de cette base comme les lignes de G.

Exemple : Soit

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

Pour construire G on résout le système de trois équations homogènes en sept inconnus:

$$\begin{aligned} x_1 + x_3 + x_5 + x_t &= 0 \\ x_2 + x_3 + x_6 + x_7 &= 0 \\ x_4 + x_5 + x_6 + x_7 &= 0 \end{aligned}$$

Par exemple, si on pose $x_5 = 1$ et $x_3 = x_6 = x_7 = 0$ on obtient $x_1 = x_4 = 1$ et $x_2 = 0$, ce qui donne le vecteur de base (1001100). On trouve les autres trois vecteurs de la base de la même façon, et on obtient

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

Afin de construire les mots de C on prend $W = (\mathbf{Z}/2\mathbf{Z})^4$ ($k = 4$) et on calcule wG pour tout $w \in W$. Le code qui en résulte contient 16 mots de longueur $n = 7$. Il s'agit d'un code parfait 1–correcteur. C'est un exemple d'un code de Hamming.
Supposons que le mot $c \in C$ est transmis et on reçoit le vecteur $r \in (\mathbf{Z}/2\mathbf{Z})^n$. Alors $r = c + e$ pour un vecteur erreur $e \in (\mathbf{Z}/2\mathbf{Z})^n$ qui contient des 1 dans les positions où r et c diffèrent et des zéros ailleurs. On note que $Hr^t = Hc^t + He^t = He^t$, donc on peut déterminer He^t en calculant Hr^t. Si on peut trouver e à partir de He^t on peut alors trouver le mot corrigé $c = r + e$. On revient à ce problème dans la suite.

La pratique : En changeant de base on écrit H sous la forme:

$$\widetilde{H} = (I_{n-k} \mid P)$$

Soit $\widetilde{G} = (-P^t | I_k)$. Alors $\widetilde{H}\widetilde{G}^t = 0$. Pour obtenir G on écrit $\widetilde{G}$ dans la base originale.

Dans l'exemple ci-dessus, il suffit d'interchanger les colonnes 3 et 4 (ce qui correspond au changement de base $e_3 \leftrightarrow e_4$), puis 1 et 3 pour obtenir $\widetilde{H}$:

$$\widetilde{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Puis

$$\widetilde{G} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Enfin, on interchange les colonnes 1 et 3, puis 3 et 4 pour obtenir G :

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

Codes de Hamming : Le code dans l'exemple ci-dessus s'appelle un code de Hamming à cause de la forme de la matrice de contrôle H. En effet, les colonnes de cette matrice sont les nombres $1, 2, \ldots, 7$ exprîmés en notation binaire. Par exemple, la colonne six est $(110)^t$, ce qui correspond à $6 = 1 \times 2^2 + 1 \times 2^2 + 0 \times 2^0$.

Plus généralement, un code de Hamming se déduit d'une matrice de contrôle H dont les colonnes sont les nombres $1, 2, \ldots, 2^m - 1$ exprîmés en notation binaire, pour un entier $m > 1$. Il s'ensuit que H est une $m \times (2^m - 1)$ – matrice dont les colonnes donnent tous les vecteurs non-nuls de longueur m sur $\mathbf{Z}/2\mathbf{Z}$. A partir de H on obtient une matrice génératrice G de taille $(2^m - 1 - m) \times (2^m - 1)$ sur $\mathbf{Z}/2\mathbf{Z}$ en trouvant une base pour le noyau de H. Enfin on construit les mots de C en calculant wG pour tous les vecteurs w de longueur $2^m - 1 - m$ sur $\mathbf{Z}/2\mathbf{Z}$. C contient $n = 2^m - 1$ mots chaqu'un de longueur $k = 2^m - 1 - m$. On parle d'un $[n, k]$–code de Hamming.

Poids de Hamming : Soit C un code linéaire construit à partir d'une matrice génératrice G et soit $x \in C$. On définit le *poids de Hamming de* x comme le nombre de fois que 1 apparait dans x : $\omega(x) := d(x, 0)$, où 0 est le mot avec zéro dans chaque composante. Soit $\omega := \min\{\omega(x) | x \in C, x \neq 0\}$. Alors $\omega = d(C)$, car, d'un part $\omega = \omega(c) = d(c, 0)$ pour un $c \in C$, d'où $d(C) \leq \omega$. D'autre part, $d(C) = d(x, y) = \omega(x - y)$ pour des $x, y \in C$. Puisque C est un espace vectoriel, alors $x - y \in C$ et $\omega \leq d(C)$.

Théorème : Soit C un code linéaire avec matrice de contrôle H et soit s le nombre minimal des colonnes linéairement dépendantes dans H. Alors $s = d(C)$.

Preuve : On définit ω comme ci-dessus. Supposons les colonnes $C_{i_1}, \ldots, C_{i_s}$ de H sont linéairement déndantes. Alors il existe $a_1, \ldots, a_s \in \mathbf{Z}/2\mathbf{Z}$ tel que

$$a_1 C_{i_1} + \cdots + a_s C_{i_s} = 0\,.$$

On considère un vecteur x dans $(\mathbf{Z}/2\mathbf{Z})^n$ qui contient a_j en position j pour $j = 1, \ldots, s$ et des zéros ailleurs. Alors $Hx^t = 0$ et donc $x \in C$. Il s'ensuit que $s \geq \omega = d(C)$.

Reciproquement, soit $y \in C$ tel que $\omega(y) = d(C)$ et soit $i_1, \dots i_d$ les positions dans y non-nulles. alors

$$0 = Hy^t = C_{i_1} + \cdots + C_{i_d}\,.$$

et les colonnes $C_{i_1}, \dots, C_{i_d}$ sont dépendantes et par suite $s \leq d(C)$.

Corollaire : Un code de Hamming est 1-correcteur.

Preuve : On voit que les trois premières colonnes de la matrice de contrôle sont dépendantes. En plus, on ne peut pas avoir deux colonnes dépendantes sinon elles seront égales ou l'une serait le vecteur 0. Par le théorème, $d(C) = 3$ et C est 1-correcteur.

Exercice : Montrer que les $[2^m - 1, 2^m - 1 - m]$–codes de Hamming sont parfaits.

Décodage pour un code de Hamming : Ce sont des codes 1-correcteur et parfaits, donc les seuls erreurs on doit considèrer sont les vecteurs d'erreur contenant des zéros sauf dans une seule position i où il y a le nombre 1. On suppose C un $[2^m-1, 2^m-1-m]$–code de Hamming et qu'on reçoit le vecteur $r \in (\mathbf{Z}/2\mathbf{Z})^{2^m-1}$. Si $r \notin C$ et puisque les colonnes de la matrice de contrôle H consistent de tous les vecteurs non-nuls de longueur m sur $\mathbf{Z}/2\mathbf{Z}$, Hr^t est une des colonnes de H, disons la colonne j. Mais cette colonne est aussi He_j^t (e_j le vecteur avec des zéros sauf dans la position j), et $Hr^t = He_j^t$. Donc l'erreur en r est e_j.

Exemple : Soit C le $[7, 4]$–code de Hamming et supposons qu'un mot est transmis et on reçoit le vecteur $r = (1011001)$. En prenant la matrice de contrôle :

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

calculer Hr^t, puis corriger r.

Décodage par syndrome : Soit C un code linéaire t-correcteur dans $(\mathbf{Z}/2\mathbf{Z})^n$. La relation

$$x \sim y \quad \Leftrightarrow \quad x - y = x + y \in C$$

est une relation d'équivalence sur $(\mathbf{Z}/2\mathbf{Z})^n$ (exercice). Si C est de dimension k, il y a 2^{n-k} classes d'équivalences.

On suppose que le mot c est transmis et qu'on reçoit le vecteur $r \in (\mathbf{Z}/2\mathbf{Z})^n$ avec $r = c + e$ pour un vecteur non-nul e (l'erreur). Puisque la différence entre r et e est un élément de C, alors r et e appartiennent à la même classe d'équivalence. Donc si

r contient $\leq t$ erreurs, on peut trouver e en cherchant le vecteur unique dans la classe de r de poids minimum (avec le plus petit nombre de 1). Dans un code avec un grand nombre de mots il n'est pas pratique de construire tous les éléments d'une classe. Mais on a la théorème suivant (exercice) :

Théorème Soit C un code linéaire avec matrice de contrôle H. Alors u et v appartiennent à la même classe d'équivalence si et seulement si $Hu^t = Hv^t$.

Le vecteur Hu^t est appelè *syndrome de* u. On peut alors trouver e en identifiant le vecteur unique avec le même syndrome de r qui contient le nombre 1, $\leq t$ fois. Si r contient plus que t erreurs, alors le syndrome de r ne correspond à aucun vecteur dans $(\mathbf{Z}/2\mathbf{Z})^n$ avec le nombre ℓ de 1 vérifiant $\ell \leq t$.

Exemple : Soit $W = (\mathbf{Z}/2\mathbf{Z})^2 = \{(00), (10), (01), (11)\}$ et soit G la matrice génératrice

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Le code qui en résulte est $C = \{(00000), (11100), (00111), (11011)\}$. Il s'agit d'un sous-espace de dimension 2 de $(\mathbf{Z}/2\mathbf{Z})^5$. Une matrice de contrôle est donnée par

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Ce code est 1–correcteur. Il s'ensuit que les vecteurs de poids minimum dans chaque classe d'équivalence sont le vecteur nul (00000) et les vecteurs qui contiennent un seul 1 : $(10000), (01000), \ldots$. On construit un tableau

leader du coset	syndrome
(00000)	$(000)^t$
(10000)	$(111)^t$
(01000)	$(100)^t$
(00100)	$(011)^t$
(00010)	$(010)^t$
(00001)	$(001)^t$

Exemple : Supposons que le mot $c \in C$ est transmis et on reçoit le vecteur $r_1 = (00011) \in {\mathbb{F}_2}^5$. Afin de corriger ce vecteur on calcule $Hr_1^t = (011)^t$. Puisque le vecteur de poids minimum (00100) a ce même syndrome, on conclut que l'erreur en r est $e = (00100)$. On corrige r_1 en $c = r_1 + e = (00111)$.

Puisque chaque classe contient quatre vecteurs, seulement 24 des 32 vecteurs dans $\mathbb{F}_1^{11}$ sont dans des classes avec un vecteur unique des poids ≤ 1. Si on a reçu $r_2 = (01001)$, on calcule $Hr_2{}^t = (101)^t$. Mais aucun vecteur de poids ≤ 1 a le même syndrome et donc r_2 n'est pas corrigible.

Tableau standard de décodage : Il s'agit d'un tableau contenant tous les mots de $(\mathbf{Z}/2\mathbf{Z})^n$. Sur chaque ligne sont rangés tous les éléments d'une même classe d'équivalence.

On procède de la façon suivante :

• première ligne : on liste les mots de C, en commençant par 0 ;

• deuxième ligne : on choisit un mot a de poids minimum qui n'est pas déjà dans le tableau (en parcourant tous les mots de poids 1, puis 2, ...). On remplit alors la ligne en inscrivant $a + x$ dans la colonne ayant au sommet le mot du code x ;

• troisième ligne : on choisit un mot b de poids minimum qui n'est pas dans le tableau et on remplit la ligne en inscrivant $b + x$ dans la colonne ayant au sommet le mot du code x ;

• on continue ainsi.

Exemple : Soit C le code linéaire de taille $(n = 4, k = 2)$ de matrice génératrice

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

c'est à dire $C = \{(0000), (1011), (0101), (1110)\}$. On obtient le tableau:

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Utilisation du tableau : Soit r le mot reçu. On cherche sa position dans le tableau, puis on le corrige par le mot c en haut de la même colonne. Cela revient à ajouter à r le vecteur d'erreur e situé en tête de sa ligne. Par constuction, e est un élément de poids minimum dans sa classe d'équivalence et par suite c est bien un mot du code le plus proche de r.

Remarques : Les codes de Hamming sont linéaires et parfaits. Pourtant, si plus d'une erreur apparaît dans le vecteur reçu, et puisque les codes de Hamming sont 1-correcteur,

on ne peut pas corriger ce vecteur. En effet, pusique le code est parfait le vecteur est uniquement corrigible mais pas au bon mot !

Dans le prochain chapitre on va étudier une classe de codes appelés codes BCH qui sont linéaires et qui sont capables de corriger plusieurs erreurs. Ces codes sont basés sur des polynômes plutot que sur des vecteurs.

Référence : R. Klima, N. Sigmon et E. Stitzinger, Applications of Abstract Algebra with Maple, CRC Press 1999.