CODES CORRECTEURS

D'ERREURS

Marc URO

TABLE DES MATIÈRES

CODES CORRECTEURS D'ERREURS

Partant d'un alphabet binaire, il peut s'avérer plus pratique de représenter les données issues d'une source binaire par des symboles q-aires, surtout lorsque q est une puissance de 2. Un symbole q-aire correspondra à un mot constitué de q éléments binaires. On considérera dans tout ce qui suit que l'alphabet de référence utilisé pour la construction du code est un alphabet de taille q.

Le principe de construction d'un code correcteur d'erreurs systématique (donc détecteur d'erreurs) consiste à ajouter aux mots constitués de m éléments q-aires d'information $a_1 a_2 \ldots a_m$, k éléments q-aires de contrôle (ou de redondance) $a_{m+1} \ldots a_{m+k}$ déterminés par le biais d'une fonction Ψ des m éléments q-aires d'information, définie au préalable.La longueur d'un mot code est alors $n = m + k$. Pour vérifier qu'un mot reçu $a_1 a_2 \ldots a_m a_{m+1} \ldots a_{m+k}$ appartient au code, on applique la fonction Ψ à $a_1 a_2 \ldots a_m$: on obtient $a'_{m+1} \ldots a'_{m+k}$. Ensuite on compare cette grandeur aux éléments q-aires effectivement reçus $a_{m+1} \ldots a_{m+k}$. S'il y a coïncidence entre ces deux grandeurs, le mot reçu est un mot code; Sinon, on détecte une erreur (puisque le mot reçu n'est pas un mot code, il ne peut avoir été émis).

Après avoir introduit quelques notions caractéristiques d'un code, nous serons en mesure de savoir le nombre d'erreurs qu'il peut **détecter** et le nombre d'erreurs qu'il peut **corriger**.

Si q est de la forme $q = p^r$ avec p nombre premier et r entier strictement positif (ce qui est le cas pour notre étude, puisque $p = 2$), on sait qu'il existe un corps fini noté GF(q) (GF signifie Galois Field, champ de Galois) constitué de q éléments. Une représentation de ce corps est obtenue en considérant les polynômes de degré inférieur ou égal à $r - 1$ et à coefficients dans le corps fini $GF(2) = \{0, 1\}$ (muni de l'addition modulo 2).

L'ensemble V de tous les mots possibles constitués de n éléments q-aires est un espace vectoriel sur GF(q). Son cardinal est q^n et sa dimension est n. Considérons alors l'ensemble C des mots code (inclus dans V): Son cardinal est q^m (les q^m mots possibles résultant de la concaténation de m éléments q-aires).

PRINCIPE DU DÉCODAGE À DISTANCE MINIMUM

Le principe du décodage à distance minimum consiste à associer au mot reçu y le mot code c le plus "proche" de y en ce sens que y et c diffèrent en un nombre de rangs le plus petit possible. Nous allons montrer que cette stratégie de décodage revient à appliquer le critère du maximum de vraisemblance. Supposons que les mots code soient transmis sur un canal binaire symétrique de probabilité d'erreur p $\left(\text{avec } p < \frac{1}{2}\right)$. Cette hypothèse est assez générale car ellle correspond au modèle du canal de transmission à bruit additif gaussien centré.

Alors si un mot code c est transformé en un mot reçu y qui diffère de c en l places, on a $P\{y/c\} = p^l (1-p)^{n-l}$. Le nombre de rangs (l) en lesquels y et c diffèrent est appelé la **distance de Hamming** et est notée $d_H(y,c)$.

Et $P\{y/c\} = g(l)$ est une fonction décroissante de l. En effet:

$g(l) = e^{l\,Lnp} e^{(n-l)Ln(1-p)}$ soit $g'(l) = (Ln\,p) e^{l\,Lnp} e^{(n-l)Ln(1-p)} + e^{l\,Lnp}\left(-Ln(1-p)\right) e^{(n-l)Ln(1-p)}$

D'où $Signe\left(g'(l)\right) = Signe\left(Ln\,p - Ln(1-p)\right)$

Si $p < \frac{1}{2}$ alors $p < 1 - p$ et $Ln\,p < Ln(1-p)$ d' où $g'(l) < 0$

Satisfaire le critère du maximum de vraisemblance en recherchant le mot code c qui rend l'observation y la plus vraisemblable revient donc à attribuer à y le mot code c le plus près de y. Le décodage de y consistera donc à trouver le mot code c de telle sorte à ce que $z = y + c$ ait le poids (nombre de "1") le plus petit possible.

Ce principe de décodage justifie les notions introduites ci-après.

On dit que le code **C est linéaire** si C est un sous-espace vectoriel de V. Dans tout ce qui suit, on se placera sous cette hypothèse.

Le **poids** d'un vecteur v de V est défini par:

$W(v) = \{\text{nombre de composantes de v différentes de } 0\}$.

À partir de cette notion, on introduit la **distance de Hamming** entre u et v:

$d(u,v) = \{\text{nombre d' éléments q - aires de même rang qui diffèrent de u à v}\}$.

Une grandeur importante est la **distance minimum** d'un code

INCORPORER "Equation" "Objet de Word15" * mergeformat

$$d_m = \inf_{\substack{u,v \in C \\ u \neq v}} d(u,v)$$

Remarque:

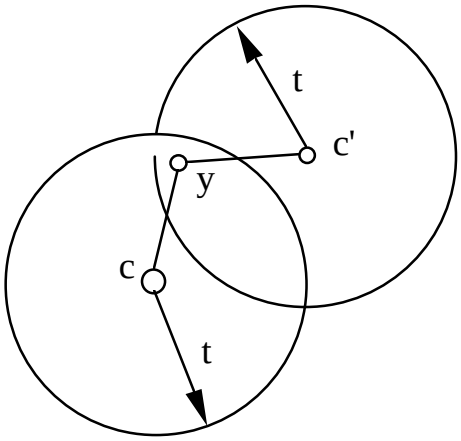
Dans le cas particulier où $q = 2$, on a $d(u,v) = W(u+v)$ et

$$d_m = \inf_{\substack{u,v \in C \\ u \neq v}} W(u+v) = \inf_{\substack{u \in C \\ u \neq 0}} W(u)$$

Proposition

Soit C un code linéaire de distance minimum d_m. Alors les boules fermées centrées sur les mots code et de rayon t sont disjointes deux à deux si $t \leq ent\left(\frac{d_m - 1}{2}\right)$ (ent(x) désigne la partie entière de x).

Raisonnons par l'absurde en supposant qu'il existe un mot reçu y et deux mots code c et c' tels que $d(y,c) \leq t$ et $d(y,c') \leq t$.



On a $d(c,c') \leq d(c,y) + d(y,c') \leq 2t$

soit si d_m est pair ie $d_m = 2p \quad t = ent\left(\frac{2p-1}{2}\right) = p - 1 \quad \Rightarrow \quad 2t = d_m - 2$

si d_m est impair ie $d_m = 2p+1 \quad t = ent\left(\frac{2p}{2}\right) = p \quad \Rightarrow \quad 2t = d_m - 1$

On aurait donc $d(c,c') < d_m$

DÉTECTION ET CORRECTION D'ERREURS

Dans tout ce qui suit on supposera que le récepteur fonctionne sur le principe du décodage à distance minimum.

CAS D'UN CANAL SANS SYMBOLE D'EFFACEMENT

On considère un canal de transmission avec q entrées (correspondant aux q éléments q-aires de l'alphabet de référence) et q sorties.

Proposition
Un code linéaire C de distance minimum d_m permet de:

- détecter au plus $d_m - 1$ erreurs (1)
- corriger au plus $t = ent\left(\frac{d_m - 1}{2}\right)$ erreurs (2)

Le point (1) est trivial car, si moins de d_m erreurs ont été commises sur un mot code, alors le mot reçu n'est pas un mot code et on sait que des erreurs se sont produites.

Pour le point (2), on utilise le résultat de la proposition précédente. Si le mot de code c émis a été transformé en le mot reçu y comportant au plus t erreurs, alors y appartient à la boule centrée en c et de rayon t, et les boules centrées sur les mots code et de rayon t étant disjointes, y est plus près de c que de n'importe quel autre mot code. Par conséquent, le principe de décodage à distance minimum va traduire le mot reçu y en le mot code c qui avait effectivement été émis. Donc les erreurs sont corrigées. On dit que le code est t-correcteur.

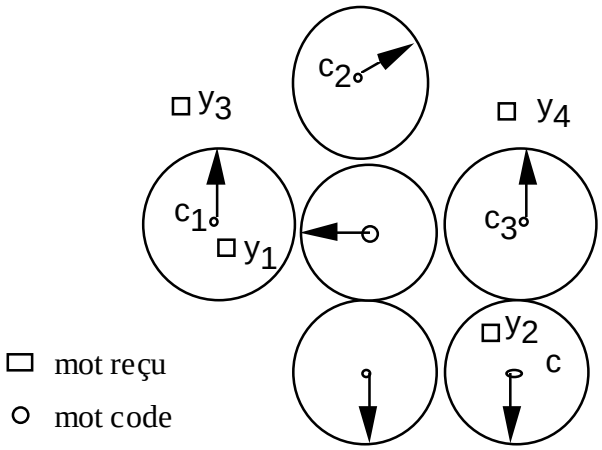
Remarque

Supposons que le mot code c_1 ait été émis. Alors on doit se trouver dans l'un des trois cas suivants:

- Le mot reçu y_1 est dans la boule centrée sur c_1 et de rayon t. Cela signifie que le nombre d'erreurs commises est inférieur à t. Le mot code choisi par le récepteur sera c_1 et les erreurs seront toutes corrigées.

- Le mot reçu y_2 est dans une boule centrée sur un mot code c (de rayon t) différent de c_1 (cela signifie que le nombre d'erreurs est supérieur à t). Le mot code choisi par le récepteur sera c et il y aura systématiquement une erreur.

- Le mot reçu y_3 n'appartient à aucune boule centrée sur un mot code et de rayon t (le nombre d'erreurs est supérieur à t). Si c_1 est le mot code situé le plus près de y, alors le récepteur choisira c_1 et les erreurs commises (en nombre supérieur à t) seront corrigées. Autrement, si le mot reçu y_4 n'appartient à aucune boule, il y aura erreur systématique
(voir illustration ci-après).



CAS D'UN CANAL AVEC SYMBOLE D'EFFACEMENT

On considère un canal à q entrées et $q+1$ sorties (dont un symbole d'effacement). Les deux propositions qui suivent vont nous montrer comment exploiter l'information apportée par la détection d'un symbole d'effacement.

Proposition
Un code linéaire C de distance minimum d_m peut "remplir" ρ effacements (c'est-à-dire remplacer dans un mot reçu les ρ symboles d'effacement par les ρ éléments q-aires effectivement émis) si $\rho \leq d_m - 1$.

Supposons que le mot code c_0 ait été émis et que ρ symboles aient été transformés en symboles d'effacement (les autres symboles constituant c_0 ont été transmis correctement).

Alors un seul mot code peut être obtenu à partir du mot reçu en remplaçant les symboles d'effacement par des symboles q-aires. En effet, s'il en existait deux, alors la distance entre ces deux mots code serait inférieure ou égale à ρ, donc strictement inférieure à d_m, ce qui est impossible. Donc le seul mot code qui peut être obtenu en remplissant les symboles d'effacement est le mot code c_0.

Proposition
Un code linéaire C de distance minimum d_m peut "remplir" ρ effacements **et** corriger t erreurs si t et ρ vérifient à la fois:
- $\rho + 1 \leq d_m$
- $2t + \rho + 1 \leq d_m$

En supprimant sur tous les mots code les éléments q-aires dont les rangs coïncident avec les rangs des symboles d'effacement apparaîssant dans le mot reçu, on obtient un nouveau code de distance minimum $d_m^{'}$ avec $d_m^{'} \geq d_m - \rho$ (le cas le plus défavorable étant celui pour lequel les rangs des ρ éléments q-aires "effacés" sont tous utilisés pour le calcul de la distance). On sait qu'alors un tel code peut corriger t erreurs si $t \leq ent\left(\frac{d_m^{'}-1}{2}\right)$.

Si la condition $2t + \rho + 1 \leq d_m$ est remplie, alors on a $2t \leq d_m - 1 - \rho$. Mais $d_m - \rho \leq d_m^{'}$ donc $2t \leq d_m^{'} - 1$, soit $t \leq \frac{d_m^{'}-1}{2}$. Comme t est entier, on a $t \leq ent\left(\frac{d_m^{'}-1}{2}\right)$ et par conséquent le code peut corriger t erreurs.

Ainsi, après correction des t erreurs, on obtient un mot code amputé de ρ symboles (remplacés par le symbole d'effacement). Comme $\rho + 1 \leq d_m$, on sait d'après la proposition précédente que le code peut remplir ces effacements.

Code parfait

On se donne un code linéaire C t correcteur et de distance minimum d_m. La longueur des mots est n, répartie en m éléments q-aires d'information et k éléments q-aires de contrôle (on a donc $n = m + k$).
On dira que le code C est **parfait** si et seulement si l'ensemble des boules fermées centrées sur les mots code et de rayon t forme une partition de l'ensemble des mots possibles en réception. En d'autres termes cela signifie que chaque mot reçu est situé dans une des boules.

Propriété
Une condition nécessaire et suffisante pour qu'un code linéaire C t-correcteur et de distance minimum d_m soit parfait est que ses paramètres vérifient la relation:
$\sum_{j=0}^{t} C_n^j (q-1)^j = q^k$

Le nombre de mots possibles en réception est: q^n. Le nombre de boules coïncide avec le nombre de mots code soit q^m.D'autre part, une boule centrée sur un mot code et de rayon t comporte:

- Le mot code c qui est le centre de la boule.
- $n(q-1)$ mots différant de c par un élément q-aire.
- $C_n^2 (q-1)^2$ mots différant de c par 2 éléments q-aires.

... ...
... ...

- $C_n^t (q-1)^t$ mots différant de c par t éléments q-aires.

Il y a donc au total $\sum_{j=0}^{t} C_n^j (q-1)^j$ mots dans une boule. Par conséquent, pour que l'ensemble des boules forme une partition de l'ensemble des mots possibles, il suffit de vérifier (on sait que les boules sont disjointes deux à deux) $q^m \sum_{j=0}^{t} C_n^j (q-1)^j = q^n$, soit: $\sum_{j=0}^{t} C_n^j (q-1)^j = q^{n-m} = q^k$ (cqfd).

Remarque

Pour un code parfait, un mot transmis avec plus de t erreurs sera systématiquement restitué de façon erronée, car le mot reçu appartiendra à une boule dont le centre n'est pas le mot code effectivement émis.

GÉNÉRATION ET DÉTECTION D'UN CODE

MATRICE GÉNÉRATRICE

Soit C un code linéaire, c'est-à dire un sous-espace vectoriel de dimension m de l'espace vectoriel V de dimension n. Alors si $(g_1, g_2, \ldots, g_m)$ est une base de C et G la matrice dont les colonnes sont les vecteurs de base de C, on peut expliciter les mots code de C:
$C = \{u \in V \,/\, u = Ga\}$ où a correspond aux vecteurs colonnes dont les composantes sont les éléments q-aires d'information. G est la matrice génératrice.

Exemples: (on se place dans le cas binaire $q = 2$)

1. On prend $n = 4$ et $m = 2$

Si on choisit pour base de C $g_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ $, g_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$, les mots code u sont obtenus à partir des quatre vecteurs a tels que:

$$u = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix} \text{ avec } (x,y) \in \{0,1\}^2$$

On obtient alors le tableau:

mot à coder	mot code	poids
00	0000	0
01	0100	1
10	1000	1
11	1100	2

On déduit $d_m = 1$. Ce code ne permet pas de détecter ni de corriger des erreurs.

2. Si on prend pour base de C $g_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$, $g_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, la matrice G devient:

$G = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$ d'où $u = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix}$ on obtient alors le code:

mot à coder	mot code	poids
00	0000	0
01	0111	3
10	1110	3
11	1001	2

La distance minimum est 2, donc le code est 1 détecteur et 0 correcteur.

Remarque

Le premier code que nous avons construit $\{0000, 0100, 1000, 1100\}$ est un code systématique. C'est-à-dire que les mots code sont constitués d'une part des bits d'information, et d'autre part des bits de contrôle. Le second code $\{0000, 0111, 1110, 1001\}$ n'est pas systématique. Le caractère systématique d'un code peut être obtenu en choisissant pour base du code des vecteurs qui soient tels que la sous-matrice obtenue en prenant les m premières lignes de G est l'identité de dimension m. Ainsi les éléments binaires de contrôle seront disposés à droite des éléments binaires d'information.

Construction du code de Hamming C(7,4) (cette notation signifie que la longueur des mots code est 7 et que le nombre d'éléments binaires d'information est 4).

On se propose de construire un code linéaire systématique parfait capable de corriger une erreur et de distance minimum minimum.

Le code étant 1 correcteur, la distance minimum doit vérifier $t = 1 = ent\left(\frac{d_m - 1}{2}\right) \Rightarrow \quad d_m = 3$ ou $d_m = 4$. Comme on s'est imposé de choisir la plus petite distance on prendra $d_m = 3$. La longueur n des mots code est donc supérieure ou égale à 3.

Le code étant parfait, on sait que ses paramètres doivent vérifier la relation $2^k = C_n^0 + C_n^1(2-1) = n+1$. Pour chaque valeur de n supérieure ou égale à 3, on va rechercher s'il existe ou non un entier k vérifiant $2^k = n+1$.

On obtient ainsi (en se limitant aux deux premières solutions trouvées):

n	k	m=n-k
3	2	1
4		
5		
6		
7	3	4

Construction du code correspondant à la première solution C(3, 1).

Comme m = 1, il n'y a que deux mots code, l'un commençant par 0, l'autre par 1. Le code étant linéaire, il constitue un espace vectoriel et par conséquent il doit contenir l'élément neutre pour l'addition qui est le mot 000. En outre la distance minimum étant 3, le second mot ne peut être que 111.

Construction du code correspondant à la seconde solution C(7, 4).
Le code étant systématique, on peut écrire les quatre premières lignes de la matrice G puiqu'on sait que cette sous-matrice de G est l'identité de dimension 4. Ensuite il reste 3 éléments à déterminer pour chacune des 4 colonnes afin d'obtenir la matrice G.

On sait qu'alors 2 au moins des trois éléments à déterminer doivent prendre la valeur 1 pour faire en sorte qu'il n'y ait pas de vecteurs de base, donc de mots code, qui aient un poids inférieur strictement à trois, sinon la distance minimum du code ne serait pas trois. De plus, il faut faire en sorte que les éléments ajoutés sur deux colonnes différentes ne coïncident pas, sinon le mot code obtenu en effectuant la somme des deux vecteurs correspondant à ces colonnes aurait un poids égal à deux, ce qui serait en contradiction avec la valeur 3 de la distance minimum.

Compte tenu de ces contraintes, on peut choisir pour G la matrice suivante:

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

En effectuant le produit de la matrice G par les vecteurs colonnes dont les quatre composantes correspondent aux éléments binaires d'information, on obtient les mots code.

mot à coder	mot code	poids
0000	0000000	0
0001	0001110	3
0010	0010101	3
0011	0011011	4
0100	0100011	3
0101	0101101	4
0110	0110110	4
0111	0111000	3
1000	1000111	3
1001	1001001	3
1010	1010010	3
1011	1011100	4
1100	1100100	3
1101	1101010	4
1110	1110001	4
1111	1111111	7

MATRICE DE CONTRÔLE

On va maintenant introduire une seconde matrice, appelée matrice de contrôle qui, appliquée au mot reçu, permettra de savoir s'il s'agit ou non d'un mot code.

On désigne par K le corps fini constitué des deux éléments 0 et 1 et C un code linéaire. C étant un sous-espace vectoriel de dimension m de l'espace vectoriel $V = K^n$ de dimension n, il possède un orthogonal que l'on notera $C^{\perp}$. La dimension de cet orthogonal est $\dim V - \dim C = n - m$. Donc à ce sous-espace $C^{\perp}$ correspond un code linéaire $(n, n-m)$ de matrice génératrice H (matrice n x (n - m)). On a alors:

$C^{\perp} = \{v \in V / v = Hb \quad \forall b \in K^{n-m}\}$

$C = \{u \in V / v^T u = 0 \quad \forall v \in C^{\perp}\}$ (v^T désigne le vecteur transposé de v)

$C = \{u \in V / (Hb)^T u = 0 \quad \forall b \in K^{n-m}\}$ soit encore:

$C = \{u \in V / b^T H^T u = 0 \quad \forall b \in K^{n-m}\}$

D'où finalement $C = \{u \in V / H^T u = 0\} = Ker\{H^T\}$.

H^T est la matrice de contrôle du code C. Pour savoir si un mot u est un mot code, il suffit de lui appliquer la matrice H^T ; Si le produit $H^T u$ est nul, on déduit que u est un mot code, sinon on sait qu'au moins une erreur a été commise.

Applications

- La matrice de controle d'un code pourra être utilisée pour surveiller la qualité d'une liaison en détectant les configurations interdites du code,

- une deuxième utilisation peut être envisagée pour corriger les erreurs en interprétant les mots reçus en des mots code, comme nous l'allons voir ci-après.

EXPRESSION DE H EN FONCTION DE G.

Soit C un code linéaire dont les mots de longueur n contiennent m éléments binaires d'information (donc n - m éléments binaires de contrôle). L'orthogonal de C noté $C^{\perp}$ est caractérisé par une matrice génératrice H.

Si le code C est systématique, les éléments binaires d'information figurant au début du mot (à gauche), alors la matrice génératrice G de C a la forme:

$$G = \begin{pmatrix} I_{dm} \\ P \end{pmatrix} \text{ avec } \begin{cases} I_{dm} = \text{matrice identité de dimension m} \times \text{m} \\ \text{P = matrice de dimension (n - m)} \times \text{m} \end{cases}.$$

La transposée de G s'écrit donc $G^T = \begin{pmatrix} I_{dm} & P^T \end{pmatrix}$
avec P^T = matrice transposée de P de dimension $\text{m} \times (\text{n - m})$.

On va montrer que la matrice H a alors la forme:

$$H = \begin{pmatrix} -P^T \\ I_{d(n-m)} \end{pmatrix}$$

En effet $\forall u \in C^{\perp}$, on doit avoir $u = Hb$ avec $b \in \{0,1\}^{n-m}$ et de la même façon:
$\forall c \in C$, $c = Ga$ avec $a \in \{0,1\}^m$.
De plus, C et $C^{\perp}$ étant orthogonaux, on doit vérifier $\langle u,c \rangle = 0 \ \forall u \in C^{\perp}$ et $\forall c \in C$. C'est-à-dire $\forall a \in \{0,1\}^m, \forall b \in \{0,1\}^{n-m} \quad (Hb)^T . Ga = 0,$ soit $b^T H^T Ga = 0$.

Il faut donc satisfaire $H^T G = 0$ ou encore $G^T \text{H} = 0$.

Si on note p_{ij} le terme général de P, on a: $G^T = \begin{pmatrix} 1 & 0 & 0 & 0 & p_{11} & p_{21} & \cdots & p_{n-m1} \\ 0 & 1 & 0 & 0 & p_{12} & p_{22} & \cdots & p_{n-m2} \\ 0 & 0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 & p_{1m} & p_{2m} & \cdots & p_{n-mm} \end{pmatrix}$

Calculons le produit $G^T H$:

$$G^T H = \begin{pmatrix} 1 & 0 & 0 & 0 & p_{11} & p_{21} & \cdots & p_{n-m1} \\ 0 & 1 & 0 & 0 & p_{12} & p_{22} & \cdots & p_{n-m2} \\ 0 & 0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 & p_{1m} & p_{2m} & \cdots & p_{n-mm} \end{pmatrix} \times \begin{pmatrix} -p_{11} & -p_{21} & \cdots & -p_{n-m1} \\ -p_{12} & -p_{22} & \cdots & -p_{n-m2} \\ \cdots & \cdots & \cdots & \cdots \\ -p_{1m} & -p_{2m} & \cdots & -p_{n-mm} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (0)$$

On a donc vérifié que $H = \begin{pmatrix} -P^T \\ I_{d(n-m)} \end{pmatrix}$.

NOTION DE SYNDROME

On a vu que si c est un mot code, alors $H^T c = 0$. Plus généralement si y est un mot reçu, c'est-à-dire un élément de $\{0,1\}^n$, on définit le **syndrome** de y par la quantité $s(y) = H^T y$. Ainsi, si un mot code c est transformé en le mot reçu y tel que $y = c + e$, cela signifie que e comporte des "1" là où une erreur a été commise. Le syndrome de y s'écrit alors: $s(y) = s(c+e) = H^T(c+e) = H^T c + H^T e = H^T e$. On constate donc que le syndrome ne dépend que de l'erreur (maladie) mais pas du mot reçu (le patient). Cette propriété va permettre de diminuer la complexité du décodage à distance minimum.

La procédure de décodage du mot reçu y se résume de la manière suivante;

On calcule le syndrome de y, $s(y) = H^T y$.

- Si $s(y) = 0$, y est un mot code et on interprète y en le mot code y.
- Sinon on recherche la séquence z de longueur n de poids minimum telle que $H^T z = s(y)$ et on décode y en $c = y + z$.

CONSTRUCTION DE LA TABLE DE DÉCODAGE

Le problème consiste à trouver z de poids minimum vérifiant $H^T z = s(y)$. Pour ce faire on utilise une table de décodage construite comme suit:
On liste les $q^{(n-m)}$ valeurs possibles s du syndrome et on recherche pour chacune la séquence z de longueur n de poids minimum vérifiant $H^T z = s$.
Le plus simple est de commencer par $z = 0$ puis de faire correspondre une valeur s aux z de poids 1, puis aux z de poids 2, ... On s'arrête lorsque toutes les valeurs s ont un z correspondant.
Comme on choisit le mot code c tel que $c = y + z$, les erreurs corrigées sont celles apparaîssant dans la table. En conséquence la probabilité d'erreur liée à cette stratégie de décodage est la probabilité des séquences d'erreurs qui ne figurent pas dans la table.

Exemple

On considère le code:

mots d'information	mots code
000	000000
001	001110
010	010101
011	011011
100	100011
101	101101
110	110110
111	111000

On note $u_1u_2u_3$ les mots d'information et $a_1a_2...a_6$ les mots code.

On a $a_1 = u_1, a_2 = u_2, a_3 = u_3$, donc le code est systématique. D'autre part les éléments binaires de contrôle s'expriment comme combinaisons linéaires des éléments binaires d'information: $a_4 = u_2 + u_3 \quad a_5 = u_1 + u_3 \quad a_6 = u_1 + u_2$ .Ceci nous permet d'écrire la matrice génératrice G du code:

$$G = \begin{pmatrix} I_{d3} \\ P \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

On déduit alors la matrice H:

$$H = \begin{pmatrix} -P^T \\ I_{d(6-3)} \end{pmatrix} = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \quad H^T = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Pour construire la table de décodage, on doit recenser les valeurs possibles des syndromes. La matrice H^T étant de dimension 3×6 et les séquences z de dimension 6×1, les produits $H^T z$ sont de dimension 3×1. Donc il y a 2^3 valeurs possibles des syndromes.

La séquence $z=(000000)^T$ a pour syndrome $(000)^T$,

la séquence $z=(000001)^T$ a pour syndrome $(001)^T$

la séquence $z=(000010)^T$ a pour syndrome $(010)^T$

etc...

Finalement on obtient:

syndromes (transposés)	séquences z (transposées)
000	000000
001	000001
010	000010
011	100000
100	000100
101	010000
110	001000
111	100100

D'après cette table de décodage, on constate que le code permet de corriger toutes les configurations de une erreur et une configuration de deux erreurs.

Supposons que l'on reçoive le mot y = 110111. Le calcul du syndrome de y défini par $H^T z=s(y)$ conduit à la valeur $(001)^T$. La table de décodage permet de déterminer $z=(000001)^T$. Le décodage de y est $c=y+z=110110$.

Si p est la probabilité d'erreur du canal binaire symétrique utilisé pour transmettre ces mots code, alors la probabilité de mauvaise interprétation d'un mot reçu est:
$1-\left\{(1-p)^6+6p(1-p)^5+p^2(1-p)^4\right\}$