Chapitre 4
Codes détecteurs d'erreurs et correcteurs d'erreurs

4. 1. Introduction

Les **codes correcteurs** ont leurs sources dans un problème de la transmission de données. Parfois, une transmission de données se fait en utilisant une voie de communication non entièrement fiable. L'objectif d'un code correcteur est l'apport d'une redondance de l'information de telle manière à ce que l'erreur puisse être détectée voire corrigée.

4.2. Correction d'Erreur et Capacité de Détection

4.2.1. Distance de Hamming

Définition : Soit *A* un alphabet et *F* l'ensemble des suites de longueur *n* à valeur dans *A*. La **distance de Hamming** entre deux éléments *a* et *b* de *F* est le nombre d'éléments de l'ensemble des images de *a* qui diffèrent de celle de *b*.

– La distance de Hamming entre a et b est

$$d_H(a,b) = \#\ \{i \mid 0 \leq i \leq n-1,\ a_i \neq b_i\}.$$

Exemple : Calculer la distance de Hamming entre a et b:
a=(0 0 0 1 1 1) et b=(1 1 0 1 0 1)
$d_H(a,b)$ = ?

Propriétés

La distance de Hamming est une distance au sens mathématique du terme:

$\forall$ a, b $\in$ F : d(a,b) = d(b,a) (symétrie)

$\forall$ a, b $\in$ F : d(a,b) = 0 $\Leftrightarrow$ a = b (séparation)

$\forall$ a, b, c $\in$ F : d(a,c) $\leq$ d(a,b) + d(b,c) (inégalité triangulaire)

On appelle **espace de Hamming** sur A l'ensemble A^n muni de la métrique d_H.

– Si A est un groupe, le poids de Hamming $w_H(x)$ d'un mot $x \in A^n$ est le nombre de ses coordonnées non nulles : $w_H(x) = d_H(x,0)$,

où 0 est le mot de A^n ayant toutes ses coordonnées égales à l'élément neutre de A.

(Un groupe est un ensemble muni d'une loi de composition interne associative admettant un élément neutre et, pour chaque élément de l'ensemble, un élément symétrique.)

4.2.2. Poids d'un mot

Définition : Le poids d'un mot est le nombre de symboles non nuls qu'il contient.
Exemple : Le poids de 10110 est 3, tandis que le poids de00000000 est 0 et que le poids de 001000 est 1.

Propriété: La distance de Hamming entre deux mots de code est le poids de leur différence : $d(z_i,z_j)=w(z_i-z_j)$

Définition: Le mot de code nul est le mot de code constitué uniquement de zéros. Il sera désigné par 0.

Propriété: Pour tous mots de code z_i et z_j, on a: $w(z_i)+w(z_j)\geq w(z_i+z_j)$

Exemple: Dans le cas binaire, $w(110101)+w(010101)=4+3=7\geq 1=w(100000)=w(110101+010101)$

4.2.3. Décodage à distance minimale et à vraisemblance maximale

Définition : On dit qu'un code C utilise un Décodage à distance minimale chaque fois que la décision de décodage D consiste, pour un mot â reçu, à choisir le (un des) mot(s) de code le(s) plus proche(s):

$$D(\hat{a},a)= \underset{a\in C}{\text{Argmin}}\, d(a,\hat{a})$$

Exemple : si les deux seuls mots de code possibles sont 000 et111, et que 010 est reçu, nous allons certainement le décoder en 000.

Définition (Distance Minimale d'un Code) : La Distance Minimale $d_{min}(C)$ d'un code $C = \{z_1,...,z_i,...,z_m\}$ est la distance de Hamming minimale (non nulle) entre toute paire de ses mots de code:

$$d_{min}(C)= \min_{i\neq j} d(z_i,z_j).$$

4.2.4. Correction d'Erreur et Capacité de Détection

Théorème: Un code par bloc de longueur n utilisant le décodage à distance minimale peut, pour toute paire d'entiers t et s tels que $0\leq t\leq n$ et $0\leq s\leq n-t$, corriger tous les schémas à t erreurs ou moins et détecter tous les schémas à t+1,...,t+s erreurs si et seulement si sa distance minimale est strictement supérieure à 2t+s.

$d_{min}(C) > 2t+s \Leftrightarrow$ C corrige t et détecte t+s erreurs.

4. 3. Les codes linéaires

5.3.1. Les codes en bloc

Définition: On appelle **code en bloc** une application C(n,k) (avec n > k) de $\{0,1\}^k$ dans $\{0,1\}^n$, qui à tout vecteur de k éléments binaires $m = [m_1,...,m_k]$ associe un vecteur de n éléments binaires $c = [c_1,...,c_n]$.

Le rendement d'un tel code en bloc vaut donc k/n.

Définition: Un code (n,k) est dit **systématique** si et seulement si il laisse intactes les k lettres d'entrée et se contente de rajouter n − k lettres supplémentaires, appelées lettres redondantes ou parity checks.

Exemples

a. Le bit de parité : sur un octet, le 8ème bit est la somme des 7 précédents. Cela donne un code (8,7) (binaire), qui permet de détecter une erreur, c'est à dire un bit changé, mais pas de la corriger.

b. Le code à répétition, envoie n fois le même bit. C'est donc un code (n,1), qui permet de détecter jusqu'à n−1 erreurs, et d'en corriger (n−1) / 2 si n est impair, par décision à la majorité. Le code à répétition offre un bon pouvoir de correction, mais c'est au prix d'un rendement très faible.

4.3. 2. Codes en bloc linéaires

4.3.2.1. Matrice génératrice

Un **code en bloc C(n,k) est linéaire** si c'est une application linéaire, c'est-à-dire, que chaque $c_{1\leq i\leq n}$ est une combinaison linéaire des $\{m_{1\leq j\leq k}\}$.
Ainsi, un code en bloc linéaire peut être défini par une matrice génératrice, souvent notée G, de dimensions k×n et pour tout mot $m \in \{0,1\}^k$, le mot de code associé est c = mG.

Par exemple, pour un code C(3,2) de matrice génératrice : $G = \begin{pmatrix} 101 \\ 011 \end{pmatrix}$

le code associé à m = [01] est c= mG = [011].

4.3.2. 2. Matrice de contrôle

L'intérêt principal des codes linéaires est qu'ils disposent de meilleurs algorithmes de décodage. On utilise pour cela les matrices de contrôle.

Définition : Soit C un code linéaire (k, n) de matrice génératrice G. On appelle matrice de contrôle de C toute matrice $H \in M_{n-k,n}$ (c'est-à-dire avec n colonnes et n − k lignes) telle que .

Théorème : (matrice de contrôle d'un code systématique)

$Hmc = \vec{0} \Leftrightarrow mc \in G$

Soit C un code systématique de matrice génératrice , G = (I_k G')

Alors la matrice $H = (G' \quad I_{n-k})$ est une matrice de contrôle de G.

Code CRC

Principe :

Basée sur des calculs de division de polynôme à coefficient dans [0, 1] –

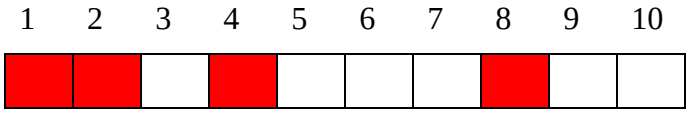
Exemple: 1 0 1 0 1 représente x4+x2+1 – Division de x3+x2+1= (x2+1)* (x + 1) + (x) : Reste =x, Quotient = x+1 – Arithmétique polynomiale modulo 2 (sans retenue): soustraction et addition sont équivalentes à un ou-exclusif bit à bit – On se donne un polynôme générateur G de degré n qui détermine le nombre de bits de contrôle

Le code de Hamming

Principe : Etant donné un bloc A de longueur : N = M + K bits, on a : M bits de message et K bits de contrôle de parité.

Exemple : Un code de Hamming 7-4 a un coefficient d'efficacité de 4/7 = 57%

- Les bits de contrôle de parité Ki sont en position 2^i pour i=0,1,2,...
- Les bits du message Mj occupent le reste du message



Les bits de données qui servent au calcul d'un bit de contrôle de numéro X sont ceux tel que X apparaît dans la décomposition en puissance de 2 de leur numéro.

Exemple: 7 = 1 + 2+ 4 donc 7 apparaît dans le calcul de 1, de 2 et de 4

Calcul des K bits de correction : les K bits de correction sont calcules en utilisant une matrice de parité H.

Exemple : Pour un code de Hamming 7-4, le calcul des 3 bits de contrôle se fait via la matrice H suivante :

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

Le bloc A envoyé est : $A = \begin{pmatrix} k1 \\ k2 \\ m1 \\ k3 \\ m2 \\ m3 \\ m4 \end{pmatrix}$ Ceci, nous donne : $H.A = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

On obtient ainsi trois équations scalaires que doivent vérifier les K bits de correction :

k1+m1+m2+m3 = 0 modulo 2

k2 + m1 +m3 +m4 = 0 modulo 2

k3 + m2 + m3 +m4 = 0 modulo 2

Le bloc A parfaitement déterminé est alors envoyé.

Réception des données et vérification :

On reçoit le bloc C = c1 c2 c3 c4 c5 c6 c7 qui peut être diffèrent du bloc A si il y a eu des perturbations sur la ligne. Si on considère qu'il n'y a eu qu'une seule erreur de transmission, alors on peut écrire :

C = A + E ou E est un bloc contenant 6 bits à 0 et 1 bit à 1.

Les positions des 0 et du 1 sont inconnues dans le bloc. On calcule le vecteur S tel que :

$$S = \begin{pmatrix} s1 \\ s2 \\ s3 \end{pmatrix} = H.C = h.(A + E) = H.A + H.E = H.E$$

Finalement, S est une des colonnes de la matrice de parité dont l'indice nous donne la position de l'erreur dans le bloc C. L'erreur est corrigée en changeant le bit considère d'état.

s3 s2 s1 est le code binaire de position de l'erreur dans le bloc C que nous obtenons à partir des équations suivantes :

s1 = (c1+c3+c5+c7) modulo 2

s2 = (c2+c3+c6+c7) modulo 2

s3 = (c4+c5+c6+c7) modulo 2

Si s3 = s2 = s1 = 0, alors il n'y a pas eu d'erreur.