Codes correcteurs

Dans la grande majorité des cas, la transmission de données se fait en utilisant un canal de communication qui n'est pas entièrement fiable : les données, lorsqu'elles circulent sur cette voie, sont susceptibles d'être altérées. Le même phénomène se produit lorsque l'on stocke de l'information sur un support. Une solution consiste à ajouter de la redondance. On obtient par exemple le code des aviateurs qui diront 'Alpha Tango Charlie' dans le but de transmettre 'ATC' à travers le radio. La séquence 'Alpha Tango Charlie', même avec friture sera plus reconnaissable pour l'oreille humaine qu'un 'ATC' déformé.

Un code correcteur à deux objectifs : détecter les erreurs et/ou corriger les erreurs. Par exemple le code correcteur utilisé pour les CDs est conçu pour corriger jusqu'à 4096 bits consécutifs, ce qui correspond à une rayure de plus d'un millimètre de large.
Ici on supposera que le message à transmettre est une suite de bits, regroupés par bloc de k bits ($k = 1$, ou 4, ou 8 ...) et que chaque bit à une probabilité non nulle d'être inversé.
Pour pouvoir corriger une éventuelle erreur dans un bloc de bits on doit nécessairement ajouter une information supplémentaire.

Exemple. Code par adjonction d'un bit de parité $(8, 9)$. On découpe notre message initial en bloc de 8 bits. On transforme ensuite chaque bloc en un bloc de 9 bits en ajoutant un bit à la fin de chaque bloc de telle sorte que la somme des bits du nouveaux blocs soit toujours 0.

Exercice 1. Coder les blocs 01101011 et 00110101. Que ce passe-t-il si un bit est modifié ? Et dans le cas de deux bits.

1. Paramètre d'un code

Un bloc de k bits sera indifféremment appelé bloc, mot ou vecteur. L'ensemble des mots de k bits est $\mathbb{F}_2^k$.
Un mot de k bits sera noté $b_1b_2...b_k$ ou éventuellement $\begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}$.

Définition 1. Un code correcteur de paramètre (k, n) est une application injective $\phi : \mathbb{F}_2^k \to \mathbb{F}_2^n$ appelée *encodage*. Le paramètre k est la *dimension* du code et n sa *longueur*.

Définition 2. Soit ϕ un code de paramètre (k, n). L'ensemble $C = \{\phi(m),\ m \in \mathbb{F}_2^k\}$ est appelé *image* du code ϕ. Les éléments de C sont les *mots de code* de ϕ.

Exercice 2. Considérons l'application $\phi : \mathbb{F}_2 \to \mathbb{F}_2^3$ définie par $\phi(0) = 000$ et $\phi(1) = 111$, c'est le code de répétition pure $(1, 3)$.

a. Préciser chacune des notions introduites pour ce code.

b. Préciser ce que peuvent devenir les mots 000 et 111 après $0, 1$ et 2 erreurs.

c. Parmi les mots trouvés, repérer ceux qui sont des mots de code.

d. Combien d'erreurs ce code peut-il détecter ? Corriger ?

Comment corriger ? Si le mot reçu n'est pas un mot de code, la probabilité qu'il se soit produit une erreur est plus importante que celle qui ne se soit produit deux erreurs. Il est donc plus raisonnable de corriger par le mot de code le plus "proche".

Définition 3. Soient m et m' deux mots de $\mathbb{F}_2^k$. On appelle *distance de Hamming* entre m et m', et on note $d(m, m')$ le nombre de bits distincts entre m et m'. On appelle *poids de Hamming* de m et on note $w(m)$ le nombre de bits non nuls de m.

Exercice 3. Montrer que pour tout m, m' de $\mathbb{F}_2^k$, on a $d(m, m') = w(m + m')$. En déduire que pour tout m, m' et c de $\mathbb{F}_2^k$ on a $d(m + c, m' + c) = d(m, m')$.

Définition 4. Soit ϕ un code d'image C. On appelle *capacité de détection* de ϕ et on note e_d le plus grand nombre d'erreurs que ϕ permet de détecter quelque soit le message. On appelle *capacité de correction* de ϕ et on note e_c le plus grand nombre d'erreurs que ϕ permet de corriger quelque soit le message. On

appelle *distance minimale* de ϕ et on note d_ϕ la plus petite distance de Hamming non nulle entre deux mots de code.

Proposition 5. *On a* $e_d = d_\phi - 1$ *et* $e_c = \left\lfloor \frac{d_\phi - 1}{2} \right\rfloor$.

Exercice 4.

a. Que vallent d_ϕ, e_d et e_c dans le cas du code bit de parité $(8, 9)$?

b. Que vallent d_ϕ, e_d et e_c dans le cas du code de répétition pure $(1, 3)$?

2. Codes linéaires

Définition 6. Un code ϕ de paramètre (n, k) est dit linéaire s'il existe une matrice $G \in M_{n,k}(\mathbb{F}_2)$ de rang k, telle que pour tout $m \in \mathbb{F}_2^k$ on ait $\phi(m) = G \times m$. La matrice G est appelée matrice génératrice du code ϕ.

Exercice 5.

a. Porqoui le rang de G soit-il être k ?

b. Les codes répétition pure $(1, 3)$ et bit de parité $(8, 9)$ sont-ils linéaires ? Si oui, quelles sont leurs matrices génératrices.

c. Etudier le code linéaire ayant

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

comme matrice génératrice.

Proposition 7. *Soit ϕ un code linéaire de paramètre (n, k). Son image C est un sous-espace vectoriel de $\mathbb{F}_2^n$.*

Proposition 8. *Soit ϕ un code linéaire d'image C. La distance minimale d_ϕ de ϕ est égale au plus petit poids non nul d'un mot de C.*

Proposition 9. *Soit ϕ un code linéaire de paramètre (k, n) et d'image C. On a $d_\phi \leqslant n - k + 1$. Un code pour lequel on a égalité est dit MDS (Maximum Distance Separable).*

Exercice 6.

a. Démontrer les 3 propositions précédentes.

b. Donner un minorant sur la longueur d'un code code linéaire de dimension k détectant d erreurs.

c. Les codes bit parité $(8, 9)$, répétition pure $(1, 3)$ et celui donnée par matrice génératrice sont-ils MDS ?

Définition 10. Soit ϕ un code linéaire de matrice génératrice G. On appelle *matrice de contrôle* de ϕ toute matrice $H \in M_{n-k,n}(\mathbb{F}_2)$ telle que $H.m = \vec{0} \Leftrightarrow m \in C$.

Définition 11. Un code ϕ de paramètre (k, n) est dit systématique si pour tout $m \in \mathbb{F}_2^k$, le mot m est un préfixe de $\phi(m)$.

Exercice 7. Montrer qu'un code de paramètre (k, n) est systématique si et seulement si sa matrice génératrice est de la forme $\begin{bmatrix} I_k \\ G' \end{bmatrix}$ où G' est une matrix de $M_{n-k,k}(\mathbb{F}_2)$.

Proposition 12. *Soit ϕ un code systématique de paramètre (k, n) et de matrice génératrice $\begin{bmatrix} I_k \\ G' \end{bmatrix}$ alors la matrice $H = \begin{bmatrix} G' & I_{n-k} \end{bmatrix}$ est une matrice de contrôle de ϕ.*

Exercice 8.

a. Démontrer cette proposition.

b. Donner la matrice de contrôle des code bit de parité $(8, 9)$, répétition pure $(1, 3)$ et de celui donné par matrice génératrice.

Définition 13. Soit ϕ un code de paramètre (k, n), de matrice génératrice G et de matrice de contrôle H. On se fixe un mot source x (de longueur k). Le mot de code correspondant sera $\phi(x) = y$. S'il y a eu des erreurs durant la transmission, on recoit z. On appelle *mot erreur* associé à z, le mot e tel que $z = y + e$. On appelle *syndrome* de z le mot Hz.

On vérifie immédiatement qu'on a $Hz = H(y + e) = He$. Le syndrome ne dépend que de la "maladie" (erreur) et non du "patient" (mot a transmettre).

Principe de décodage Soit ϕ un code linéaire de paramètre (k, n), corrigeant e_c erreurs et de matrice de contrôle H.

1. On recoit le mot z transmis avec de possibles erreurs.
2. On calcule le syndrome s de z par $s = Hz$.
3. Si s vaut 0, on ne détecte pas d'erreur et on retourne $\phi^{-1}(z)$.
4. Sinon on recherche l'erreur e de plus petit poid possible telle que $He = s$.
5. Si le poid $w(e)$ est inférieur ou égale à e_c, on retourne $\phi^{-1}(z + e)$.
6. Sinon afficher "Impossible de corriger les erreurs".

Il nous reste à voir comment calculer $\phi^{-1}(c)$ pour un mot c de $\mathbb{F}_2^n$ (utiliser au 2 et 5) et comment trouver l'erreur e (ligne 4). Si ϕ est un code systématique, alors m est le préfixe de longueur k de $\phi(m)$. Dans ce cas $\phi^{-1}(c)$ et le préfixe de longueur k de c.

3. Table de décodage

Pour trouver e, nous allons construire une *table de décodage*. Soit ϕ un code linéaire de paramètre (k, n). Une matrice de contrôle associée à ϕ est de taille $(n - k) \times n$. L'ensemble des syndromes possibles est donc $\mathbb{F}_2^{n-k}$. Pour calculer la table de décodage de ϕ, on liste tous les syndrômes de ϕ. Puis on liste tous les mots z de $\mathbb{F}_2^n$ par poids croissant. Pour chaque z, on calculer Hz. Au syndrome s on associe alors le premier mot z apparu vérifiant $Hz = s$. On arrete ce procédé dés qu'on a associé un mot à chaque syndrome. Le mot associé à un syndrome s est alors l'erreur de poid minimal donnant s.

Exercice 9.

a. Calculer la table de décodage pour le code bit de parité $(8, 9)$, répétition pure $(1, 3)$ et celui donner par matrice génératrice.

b. Pour chacun des codes respectivement, décoder les mots recus 101010100, 101 et 1100.

Exercice 10. En `Sage`, un code correcteur linéaire sera naturellement représenté par sa matrice généreatrice. Ecrire les fonctions suivantes en `Sage`

a. `image` qui retourne la liste des mots de code d'un code correcteur linéaire;

b. `distance` qui retourne la distance d_ϕ d'un code correcteur linéaire;

c. `est_mds` qui teste si le code correcteur linéaire est MDS;

d. `est_systematique` qui teste si un code correcteur linéaire est systématique;

e. `matrice_controle` qui retourne la matrice de contrôle d'un code correcteur linéaire systématique;

f. `table_decodage` qui retourne la table de décodage d'un code correcteur linéaire systématique;

g. `decode` qui étant donné un code correcteur linéaire systématique et sa table de décodage décode un mot recu.

Les fonctions précédentes seront testées avec les trois codes déjà vus ainsi qu'avec le code de Hamming $(7, 4)$ de matrice génératrice

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