Chapitre 1

Codes

1.1 Introduction

A l'ère du numérique se pose la question de la fiabilité des communications, et ce même lors de transitions dans des canaux dits bruités. Les codes correcteurs d'erreurs permettent ainsi de détecter et de corriger certaines erreurs grâce à un dispositif de codage, qui repose sur l'introduction de redondance dans le message transmis.

1.2 Codes sans structure

On considère l'ensemble E des messages et une application f *injective* appelée *encodeur* ou application de codage :

$$f : \left\{ \begin{array}{ccc} E & \to & Q^n \\ a = (a_1, ..., a_k) & \mapsto & c = (c_1, ..., c_n) \end{array} \right.$$

où Q est un ensemble à q éléments que nous appelons *alphabet* du code.

On note alors $C = f(E)$ le *code de longeur n* associé. Pour évaluer la différence entre deux mots x et y de Q^n, on définit la *distance de Hamming* définie par :

$$d(x, y) = Card\{i \in \{1, ..., n\} | x_i \neq y_i\}$$

La *distance minimale* du code C est un paramètre caractéristique du code. En considérant les boules centrées sur les mots du code, on remarque qu'un code de distance d corrige au plus $e = [\frac{d-1}{2}]$ erreurs, ce qu'on nomme la *capacité de correction* de C.

1.3 Codes linéaires

Ici on considère E comme espace vectoriel sur un corps fini, et on rappelle que ces derniers sont de la forme $\mathbb{F}_q$ où $q = p^m$ et p est premier.
On choisit naturellement l'alphabet de la forme $Q = \mathbb{F}_q$ et $E = \mathbb{F}_q^k$, $k \leq n$, ainsi que l'encodeur f linéaire, de manière à ce que C soit un sous-espace vectoriel de $\mathbb{F}_q^n$. On définit le *poids de Hamming* d'un mot de $\mathbb{F}_q^n$ par

$$w(x) = Card\{i \in \{1, ..., n\} | x_i \neq 0\} = d(x, 0)$$

le nombre de coordonnées non nulles de x. La distance minimale du code devient, grâce à la structure d'espace vectoriel de C, le poids minimal des mots de C :

$$d = \min\{w(x) | x \in C\}$$

On se place dans les bases canoniques, et on note ${}^tG$ la matrice de f, ce qui entraîne la caractérisation suivante pour c dans $\mathbb{F}_q^n$:

$$c = (c_1, ..., c_n) \in C \Leftrightarrow c = xG$$

pour un x dans $\mathbb{F}_q^k$. G est appelée *matrice génératrice* du code C.
D'autre part, toute matrice de taille $(n-k) \times n$ vérifiant :

$$c \in C \Leftrightarrow H^t c = 0$$

est appelée *matrice de contrôle* de C. On a $C = \ker H$ et $\mathrm{rg}H = n - k$.

1.4 Codes MDS - Borne de Singleton

Si C est un code linéaire $[n, k, d]$ de matrice de contrôle H, alors

$$d = \min\{s \in \mathbb{N}^* | \text{il existe s colonnes de } H \text{ linéairement dépendantes}\}$$

On en déduit la borne de Singleton : Pour tout code [n, k, d], on a $d \leq n - k + 1$.

Les codes vérifiant $d = n + k - 1$ sont appelés codes *MDS* (Maximum Distance Separable). Ils sont dans ce sens optimaux. Si on définit les *taux de correction* et *taux de transmission* ou *rendement* par $t_c = \frac{e}{n}$, $R = \frac{k}{n}$, on a une majoration intéressante :

$$2t_c + R \leq 1$$

On ne peut donc optimiser simultanément la correction et la transmission.

1.5 Codes cycliques

Un code linéaire C est dit cyclique s'il est stable par permutation à droite de ses composantes :

$$(a_0, ..., a_{n-1}, a_n) \in C \Rightarrow (a_{n-1}, a_0, ..., a_{n-2}) \in C$$

L'algèbre quotient $\mathbb{F}_q[x]/(x^n - 1)$ est un $\mathbb{F}_q$-espace vectoriel de dimension n, dont une base est $(1, x, ..., x^{n-1})$. On peut donc identifier un élément $a = (a_0, a_1, ..., a_{n-1})$ de $\mathbb{F}_q^n$ au polynôme $a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$ de $\mathbb{F}_q[x]/(x^n - 1)$. C est alors un sous-espace vectoriel de $\mathbb{F}_q^n/(x^n - 1)$, et même un idéal : il contient alors tous les $x^i a(x)$ et par stabilité, tous les $b(x)a(x)$. Réciproquement, tout idéal de cet espace est un sous-espace vectoriel stable par permutation à droite des coordonnées dans la base $(1, x, ..., x^{n-1})$ de $\mathbb{F}_q[x]/(x^n - 1)$.

En fin de compte : *Moyennant l'identification des $\mathbb{F}_q$-espaces vectoriels isomorphes $\mathbb{F}_q^n$ et $\mathbb{F}_q[x]/(x^n - 1)$, les codes cycliques sont les idéaux de $\mathbb{F}_q[x]/(x^n - 1)$.*

Or un idéal de $\mathbb{F}_q[x]/(x^n - 1)$ est monogène et engendré par la classe modulo $x^n - 1$ d'un unique polynôme unitaire g, qui divise $x^n - 1$. On l'appelle *le polynôme générateur* du code cyclique C, qui est alors de dimension $k = n - deg(g)$.

1.6 Codes BCH

Les codes de Bose-Chaudhuri-Hocquengem (BCH) permettent de construire un code cyclique avec une minoration de la capacité de correction. Pour choisir un code BCH, on se donne un entier n, une puissance d'un nombre premier q correspondant à l'alphabet $\mathbb{F}_q$ utilisé pour l'encodage, et un entier δ inférieur à n. On trouve alors β une racine primitive n^e de l'unité dans $\mathbb{F}_q$, et on définit le générateur du code :

$$g(x) = \text{ppcm}(m_{\beta^i}, ..., m_{\beta^{i+\delta-2}})$$

où m_γ est le polynôme minimal de γ sur $\mathbb{F}_q$.

Alors la distance minimale de C est supérieure à δ, appelée *distance construite* de C. Notons qu'a priori le degré de $g(x)$ peut être bien supérieur à $\delta - 1$. Comme la dimension du code k vaut $n - \deg g$, il faut être prudent pour le choix du générateur afin de conserver un taux d'information élevé. On a donc intérêt à choisir l'indice i judicieusement pour une distance construite donnée afin de maximiser de dernier, ce qui peut se faire en considérant les classes cyclotomiques modulo n sur $\mathbb{F}_q$.

1.7 Codes de Reed-Solomon

Un code de Reed-Solomon (RS) est un code BCH de longueur $n = q - 1$ sur $\mathbb{F}_q$. Dans ce cas, $x^n - 1$ est scindé à racines simples sur $\mathbb{F}_q$ puisque ses racines sont précisément tous les éléments non nuls de ce corps. On peut donc prendre $\beta = \alpha$ si α est un élément primitif de $\mathbb{F}_q$, c'est-à-dire un générateur de son groupe multiplicatif $\mathbb{F}_q^*$, et le générateur du code est exactement de degré $\delta - 1$:

$$g(x) = \prod_{i=1}^{\delta-1} (x - \alpha^i)$$

Par propriété des codes BCH, la distance d de C vérifie $d \geq \delta$, et d'après la borne de Singleton $d \leq n - k + 1 = \delta$. Donc la distance construite d'un code RS est *exactement* égale à sa distance réelle, et c'est un code MDS.
Ce sont la simplicité de construction et la performance de ce code qui justifient sa fréquente utilisation en pratique. Par exemple, la sonde *Voyager* de la NASA utilise un code RS(255, 225, 33) pour transmettre les images.

Chapitre 2

Implémentation

2.1 Construction et calculs dans les corps finis

Tout corps fini de la forme $\mathbb{F}_q$, où $q = p^m$, est une extension de son sous-corps premier $\mathbb{F}_p$ (parfois appelé corps de Frobenius). Dans l'implémentation considérée, le module *Frobenius* définit le type associé et les règles de calcul usuelles modulo p. Pour construitre le surcorps considéré, on peut prendre un polynôme $f(x)$ irréductible sur $\mathbb{F}_p$, de degré m, et voir $\mathbb{F}_q$ comme $\mathbb{F}_p[x]/(f(x))$. Les foncteurs *Polynome*, *Cantorzass* et *Cyclo* ont été prévus à cet effet, en se basant sur l'algorithme de Cantor-Zassenhaus pour la factorisation d'un polynôme sur un corps fini.

Plus précisément, cet algorithme permet ici de trouver des polynômes irréductibles dits *primitifs*, c'est-à-dire que leurs racines dans un surcorps de $\mathbb{F}_p$, ici $\mathbb{F}_q$, sont des générateurs du groupe multiplicatif (cyclique) $\mathbb{F}_q^*$. Cette propriété permet d'exploiter une autre représentation du corps, sous la forme $\mathbb{F}_q = \{0, 1, \alpha, ..., \alpha^{q-2}\}$ où α est donc un élément primitif racine de $f(x)$. Cela permet d'effectuer rapidement les multiplications d'un part, et pouvoir contrôler les racines d'un éventuel polynôme générateur d'un code BCH d'autre part.

Le foncteur *ExtensionOpt* permet ainsi de créer une extension de degré quelconque d'un corps fini pré-existant, et de jongler avec les deux représentation via une table de correspondance : l'addition et la multiplication s'effectuent ainsi en temps logarithmique en la taille du cardinal. Cependant, la constrution des tables représentées en vecteurs est rapidement inenvisageable, c'est pourquoi un module *ExtensionNonOpt* prévoit des calculs exclusivement avec une représentation polynomiale, la multiplication s'effectuant modulo $f(x)$.

2.2 Codage des codes BCH

Etant donnés n et $q = p^r$, avec n et p premiers entre eux, trouver un code BCH correspondant revient à factoriser $x^n - 1$ sur $\mathbb{F}_q$. Pour ce faire, il faut considérer une extension de la forme $\mathbb{F}_{q^m}$ où m est l'ordre multiplcatif de q modulo n, le corps de décomposition de x^n- sur $\mathbb{F}_q$ (ou corps des racines n^e de l'unité dans $\mathbb{F}_q$). On prend α générateur de $\mathbb{F}_{q^m}$ et, en posant $s = \frac{q^m-1}{n}$, $\beta = \alpha^s$ est une racine primitive n-ième de l'unité. On sait alors que les facteurs irréductibles de $x^n - 1$ sont les polynômes minimaux des puissances de β, et plus précisément que celles-ci se répartissent en classes cyclotomiques. En effet, un élément $\gamma \in \mathbb{F}_{q^m}$ est racine d'un polynôme irréductible sur $\mathbb{F}_q$ si et seulement si ses conjugués, de la forme $Conj(\gamma) = \{\gamma, \gamma^q, ..., \gamma^{q^v}\}$, le sont également.

En fin de compte, il suffit de déterminer les classes cyclotomiques sur $\mathbb{F}_q$ modulo n pour avoir la factorisation souhaitée : c'est dans un premier temps le rôle du module *Decomposition*. Ensuite, à partir des classes calculées et d'un paramètre δ, il permet de sélectionner un ensemble de polynomes irréductibles, dont le produit forme un polynôme générateur de degré minimal pour la distance construite choisie. Cela permet donc de générer un code minimisant la redondance des données pour une longueur n, un alphabet et une distance construite fixés.

2.3 Décodage des codes BCH

L'implémentation considérée utilise une technique de décodage adaptée de l'algorithme d'Euclide, qui permet de résoudre l'équation dite "clef".

Soit $g(x) = a(x)\prod_{i=b}^{b+\delta-2}(x-\beta^i)$ où $a(x)$ a pour racines le reste des conjugués de éléments considérés. Supposons qu'un message codé par un mot m, produise un mot reçu r et que le nombre l d'erreurs soit inférieur à la capacité de correction du code. Alors le mot erreur $e(x)$ est défini par $r(x) = m(x) + e(x)$. On considère le *syndrome* $S(x)$ défini par

$$S(x) = s_b + s_{b+1}x + ... + s_{b+\delta-2}x^{\delta-2}$$

où $s_j = r(\beta^j) = e(\beta^j)$, $b \leq j \leq b+\delta-2$.
On repère les coefficients non nuls de $e(x)$ par $J = \{i_1, ...i_l\}$: $e(x) = e_{i_1}x^{i_1} + ... + e_{i_l}x^{i_l}$.
On note également pour plus de clarté $X_j = \beta^{i_j}$ et $Y_j = e_{i_j}$.
On définit le *polynôme localisateur* et le *polynôme évaluateur* par :

$$\Lambda(x) = \prod_{j=1}^{l}(1-X_jx)\ ;\ \Omega(x) = \sum_{i=1}^{l} Y_iX_i^b \prod_{j=1,j\neq i}^{l}(1-X_jx)$$

Alors l'*équation-clé* s'exprime par :

$$S(x)\Lambda(x) = \Omega(x) \bmod x^{\delta-1}$$

L'algorithme d'Euclide permet alors de déterminer Λ et Ω. On cherche ensuite les racines de Λ, pour avoir la localisation des erreurs, par la recherche dite de Chien. Enfin, on peut déduire la valeur de l'amplitude des erreurs avec l'algorithme de Forney :

$$Y_j = -X_j^{1-b}\frac{\Omega(X_j^{-1})}{\Lambda'(X_j^{-1})}$$

Ces techniques sont utilisées dans le module BCH, pour le décodage.

2.4 Conclusion

La théorie des codes correcteurs est très riche et permet d'atteindre des bonnes performances en temps et en capacité de correction. Néanmoins, la mise en place de techniques efficaces pour effectuer ces opérations est parfois compliquée, et les calculs sur les corps finis sont longs dans les cas non optimisés.

2.5 Bibliographie

Papini O., Wolfmann J., *Algèbre discrète et codes correcteurs*, Springer-Verlag, **1995**

Meunier P., *Algèbre et informatique - Applications aux codes linéaires correcteurs d'erreurs*, Cépaduès, **2013**

Pretzel O., *Error-Correcting Codes and Finite Fields*, Oxford Applied Mathematics and Computing Science Series, **1996**

Augot D., Orsini E., Betti E., *An introduction to linear and cyclic codes*, Gröbner Bases, Coding, and Cryptography, Springer-Verlag, pp.47-68, **2009**