ACCQ204 4 Jan. 2016

Cours 3

Enseignant: Aslan Tchamkerten *Crédit: Pierre de Sainte Agathe*

1 Code de Hamming

Définition 1 *Pour tout entier $r \geq 2$ un code de Hamming (binaire) a pour matrice de parité H_r telle que :*

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

où la i^{eme} colonne est la représentation de i en binaire ($1 \leq i \leq 2^r - 1$) sur r bits.

Proposition 1 *Pour tout $r \geq 2$ le code de Hamming a distance minimale égale à 3.*

Preuve Les colonnes de la matrice sont deux à deux indépendantes, donc $d > 2$. De plus la somme des trois premières colonnes satisfait $H_r^1 + H_r^2 + H_r^3 = 0$, elles sont donc dépendantes et donc $d = 3$. ■

Observation 1 *Par la borne de Hamming*

$$|C| \cdot \mathrm{Vol}(n, \lfloor \frac{d-1}{2} \rfloor) \leq 2^n$$

Pour $d = 3$ on a

$$|C| \leq 2^n \cdot \frac{1}{n+1}$$

car $\mathrm{Vol}(n, 1) = n + 1$. *Il suit que*

$$\log_2(|C|) \leq n - \log_2(n+1).$$

Pour le code de Hamming, $n = 2^r - 1 \Rightarrow r = \log_2(n+1) \Rightarrow \log_2 |C| = 2^r - r - 1 = n - \log_2(n+1)$. Tout code de Hamming atteint la borne de Hamming et est donc parfait.

Observation 2 *Il existe d'autres codes parfait, par exemple, le code* $[n, 1, n]_2$*, ainsi que d'autres codes du a Golay.*

1.1 Décodage code de Hamming

1. Algo 1 : MAP. la complexité est en $2^{O(n)}$! (besoin de lister tous les mots codes.

2. Algo 2 :

   - Si le mot y reçu vérifie $Hy = 0$ alors c'est un mot code. FIN
   - Sinon, on "flip" successivement chaque bit de y et on vérifie si le mot obtenu appartient à C.

   La complexité est alors en $O(n \cdot T(n))$ où $T(n) = O(n \cdot \log_2(n))$ est la complexité du test de vérification d'appartenance (multiplication d'un vecteur par une matrice) pour une position "flipée". D'ou une complexité totale en $O(n^2 \log_2(n))$

3. Algo 3 : Dans le cas où une erreur se produit au maximum par mot code envoyé on a $y = C + e$ avec

$$e = \begin{pmatrix} 0 \\ . \\ 0 \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

   Alors $Hy = Hc + He = He$ ce qui correspond à la i^{eme} colonne de H (i étant la position du 1 dans e et donc de l'erreur dans y). La complexité est ici en $O(n \log_2(n))$ (un seul calcul matriciel à faire).

2 Codes MDS : maximum distance separable

Rappel : la borne du singleton nous dit que pour tout code

$$d \leq n - k + 1 \Rightarrow r + \delta \leq 1 \Rightarrow r \leq 1 - \delta.$$

Soit q^k mots codes formant un code de distance minimale d. En retirant $d-1$ symboles dans chaque mot code (à la même position pour tous les mots codes), on obtient q^k mots codes tous différents car ils forment un code de distance minimale ≥ 1. L'unicité des q^k mots codes du code "ponctué" donne $q^k \leq q^{n-d+1} \Rightarrow d \leq n-k+1$.

Corollaire 1 *Un code satisfait $d = n-k+1 \iff$ Pour tout ensemble de k coordonnées les mots codes sont distincts (i.e. les k composantes de tout mot code définissent le mot code).*

Définition 2 *Un code est dit MDS (maximum distance separable) si $d = n-k+1$.*

Définition 3 *Soit C un code avec q^k mots codes sur $\mathbb{F}_q$ et de longueur n. Soit J un sous-ensemble de $\{1,2,..,n\}$ de coordonnées. J est un* ensemble d'information *si pour tout mot code, les composantes de J le détermine entièrement.*

Corollaire 2 *Pour un code MDS, tout J avec $|J| = k$ est un ensemble d'information.*

Conjecture 1 *Tout code linéaire $[n,k]_q$ MDS satisfait $n \leq q+1$ si $1 < k < q$ sauf si q est pair et $k = 3$ ou $k = q-1$ auquel cas on a $n \leq q+2$.*

3 Codes de Reed-Solomon (vers 1950)

Soit $k \in [1,n], \mathbb{F}_q$ tel que $n \leq q$ et $\alpha_1, \alpha_2, ..., \alpha_n$ des *points d'évaluation* distincts de $\mathbb{F}_q$.

Soit le code

$$C = \{(f(\alpha_1), f(\alpha_2), ..., f(\alpha_n)) : f \in \mathbb{F}_q, deg(f) < k\}$$

Ce code, appelé code de Reed-Solomon, est un code linéaire car pour tout message m et m'

$$f_m(X) + f_{m'}(X) = f_{m+m'}(X)$$

et

$$a \cdot f_m(X) = f_{a \cdot m}(X)$$

Ce code a pour paramètres:

- longueur n
- dimension q^k. En effet, si il existait $f \neq f'$ telles que $f(\alpha_i) = f'(\alpha_i)\ \forall i$ et telles que $deg(f) < k$ et $deg(f') < k$, alors en posant $g = f - f'$ on aurait que le nombres de racine de g est $\geq n \geq k$ alors que $deg(g) < k$ ce qui impossible. On déduit dont que tout polynôme donne un mot code différent.
- une distance minimale $d = n - k + 1$. En effet par un raisonnement analogue on a
$$d = \min_{c \in C, c \neq 0} wt(c)$$
et comme le nombre de racines est au plus $k-1$, on a que $d \geq n-(k-1)$. Il suit que $d = n - k + 1$ par la borne supérieure de Singleton.

Observation 3 *Les codes de Reed-Solomon sont donc des codes MDS. A noter que la borne de Singleton implique la borne asymptotique $R \leq 1-\delta$. Or nous savons que la borne de Plotkin $R \leq 1-\theta$ avec $\theta = 1 - \frac{1}{q}$ est strictement meilleure et donc $R = 1 - \delta$ ne peut être atteint. Plus précisément, étant donné un alphabet $[q]$ on ne peut pas construire une suite de code de telle façon à obtenir $R = 1 - \delta$. Si l'on permet d'augmenter la taille de l'alphabet avec n, comme pour les codes RS, on peut atteindre $R = 1 - \delta$ asymptotiquement.*

Observation 4 *$RS(n, k-1) \subseteq RS(n, k)$ car les polynôme de degré $\leq k$ sont aussi de degré $\leq k - 1$.*

Observation 5 *Eliminer (ponctuer) une même coordonnée à tous les mots codes d'un code de RS(n,k) donne un code de Reed Salomon (on fait une évaluation en moins) pour autant que $n - 1 \geq k$.*

3.1 Décodage

Soit C un code RS, $(\alpha_1, \alpha_2, ..., \alpha_n) \in \mathbb{F}_q^n$, et $c \in C$ tel que $c_i = f(\alpha_i)$
On observe $y = c + e$ et l'on veut retrouver y.

CAS 1: Pas d'erreur
$y_i = f(\alpha_i)\ \forall i$.

Alors

$$\begin{pmatrix} y_1 \\ . \\ . \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & \alpha_1 & {\alpha_1}^2 & . & . & . & \alpha_1^{k-1} \\ 1 & \alpha_2 & {\alpha_2}^2 & . & . & . & \alpha_2^{k-1} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & \alpha_n^{k-1} \end{pmatrix} \cdot \begin{pmatrix} m_0 \\ . \\ . \\ m_{k-1} \end{pmatrix}$$

La matrice des alphas étant de rang plein (matrice de vandermonde) on peut retrouver le message m.

CAS 2: erreurs
On définit

$$\Lambda(X) = \prod_{j:e_j \neq 0} (X - \alpha_j)$$

comme étant le *polynôme localisateur d'erreur*. On remarque que les racines de Λ donnent les localisations des erreurs. Si l'on parvient à connaître Λ, on peut retrouver et éliminer les erreurs pour autant que leur nombre est $\leq d - 1$ (propriété MDS).

Observation 6 *Le polynôme Λ satisfait*

$$\Lambda(\alpha_i) \cdot y_i = \Lambda(\alpha_i) \cdot f(\alpha_i)$$

ou encore

$$\Lambda(\alpha_i) \cdot c_i = \Lambda(\alpha_i) \cdot f(\alpha_i)$$

car si il y a erreur en i, $\Lambda(\alpha_i) = 0$, et sinon, $y_i = f(\alpha_i) = c_i$ la ième coordonnée du vecteur envoyé.

Le problème de décodage est donc

Problème 1 *Trouver $\Lambda(X)$ et $f(X)$ tels que*

$$\Lambda(\alpha_i) \cdot (y_i - f(\alpha_i)) = 0 \quad \forall i \tag{1}$$

avec $deg(f) \leq k - 1$ et $deg(\Lambda)$ minimal.

Le difficulté est que (1) est une équation avec des termes multivariés (produits de coefficients de Λ et f) ce qui rend la solution possible mais complexe à trouver.

3.2 Relaxation du problème

Problème 2 *Trouver $\Lambda(X)$ et $f(X)$ tels que*

$$\Lambda(\alpha_i) \cdot y_i - h(\alpha_i) = 0 \quad \forall i$$

avec $deg(h) < k + deg(\Lambda)$ et $deg(\Lambda)$ minimal (on a juste remplacé le terme non linéaire $\Lambda \cdot f$ dans (1) par un terme linéaire h).

Le problème s'écrit alors :

$$\begin{pmatrix} y_1 & 0 & & \\ 0 & y_2 & & \\ 0 & 0 & . & \\ . & . & . & y_n \end{pmatrix} \begin{pmatrix} 1 & \alpha_1 & {\alpha_1}^2 & . & \alpha_1^t \\ 1 & \alpha_2 & {\alpha_2}^2 & . & \alpha_2^t \\ . & . & . & . & . \\ . & . & . & . & \alpha_n^t \end{pmatrix} \begin{pmatrix} \Lambda_0 \\ . \\ . \\ \Lambda_t \end{pmatrix} = \begin{pmatrix} 1 & \alpha_1 & {\alpha_1}^2 & . & \alpha_1^{k+t-1} \\ 1 & \alpha_2 & {\alpha_2}^2 & . & \alpha_2^{k+t-1} \\ . & . & . & . & . \\ . & . & . & . & \alpha_n^{k+t-1} \end{pmatrix} \begin{pmatrix} h_0 \\ . \\ . \\ h_{k+t-1} \end{pmatrix}$$

où $t-1$ est le degré de Λ. On essaie de résoudre pour $t = 0$, $t = 1$,... jusqu'au moment où on trouve une solution pour $\Lambda(X)$ et $h(X)$. Si cette solution est telle que $h(X) = \Lambda(X)f(X)$, i.e., si Λ divise h (ce qui est simple à vérifier) alors le problème (1) est résolu, i.e., la solution du problème relaxé est la même que celle du problème original.

1. Comment garantir qu'une solution existe? Il suffit pour cela d'avoir au moins n *degrés de liberté*. Donc il suffit que

   $$t + 1 + k + t \geq n$$

   ce qui est impliqué par la condition

   $$t \geq \frac{d-1}{2}$$

   puisque $d = n - k + 1$. Autrement dit, des que $t \geq \frac{d-1}{2}$ (la motié de la distance minimale) on a la guarantie d'avoir une solution au problème 2.

2. Cette solution est-elle valide, i.e., $h = \Lambda \cdot f$? La solution au problème 2 nous dit que Λ et f satisfont

   $$\Lambda(\alpha_i) \cdot y_i - h(\alpha_i) = 0$$

   $$\Leftrightarrow \Lambda(\alpha_i) \cdot (c_i + e_i) - h(\alpha_i) = 0 \qquad (2)$$

où e représente le vecteur d'erreur. L'observation 6 nous dit qu'on a

$$\Lambda(\alpha_i) \cdot c_i - f(\alpha_i)\Lambda(\alpha_i) = 0 \tag{3}$$

En soustrayant (3) de (2):

$$\Lambda(\alpha_i).e_i - [h(\alpha_i) - f(\alpha_i)\Lambda(\alpha_i)] = 0.$$

Définissons le vecteur $S(\cdot) = h(\cdot) - f(\cdot)\Lambda(\cdot)$. L'équation devient

$$\Lambda(\alpha_i) \cdot e_i = S(\alpha_i)$$

ce qui demande en particulier que

$$wt([\ldots \Lambda(\alpha_i) \cdot e_i \ldots]) = wt([\ldots S(\alpha_i) \ldots]).$$

Or

$$wt(\Lambda(\alpha_i).e_i) \leq t$$

et, par le théorème fondamental de l'algèbre on a que

$$wt([\ldots S(\alpha_i) \ldots]) \geq n - (k-1) - t = d - t$$

si S est non nul, puisque $deg(S) \leq k-1$. Ces deux bornes nous donnent que si $d - t > t$, i.e.,

$$\frac{d}{2} > t,$$

les poids des vecteurs ne peuvent être les mêmes ce qui est impossible, à moins que S soit le vecteur identiquement nul, i.e., $h(\alpha_i) = f(\alpha_i) \cdot h(\alpha_i)$ pour tout i.

En combinant 1. et 2. il suit que la procédure de décodage s'arrête pour un

$$t \leq \left\lceil \frac{d-1}{2} \right\rceil.$$

Si en plus

$$t < \frac{d}{2},$$

la solution trouvée est juste. Donc les codes de RS permettent de décoder correctement jusqu'a $\approx d/2$ erreurs, c'est qui est optimal. De plus le décodage est de faible complexité; la résolution du système linéaire déquation plus haut peut se faire avec complexité $O(n^3)$.