**ACCQ204** 3 Dec. 2017

*Enseignant: Aslan Tchamkerten* *Cours 3*

# 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 $i^{ème}$ colonne est la représentation de $i$ en binaire ( $1 \leq i \leq 2^r - 1$ ) sur $r$ bits. Donc $r = n - k$, i.e., $k = n - r$.*

**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 $d \geq 3$. De plus $H_r^1 + H_r^2 + H_r^3 = 0$ et donc $d = 3$. ■

**Observation 1 (Code et borne de Hamming)** *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,*

$$n = 2^r - 1 \Rightarrow r = \log_2(n+1)$$

*et donc*

$$\log_2 |C| = k = n - r = n - \log_2(n+1).$$

*On déduis que les codes de Hamming atteignent la borne de Hamming.*

**Observation 2** *Un code atteignant la borne de Hamming est dit parfait. 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 : 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 $y$).

Complexité: $O(n \log_2(n))$ (un seul calcul matriciel à faire).

Remarque: $Hy$ est appelé le syndrôme de $y$.

# 2 Codes MDS : maximum distance separable

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

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

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

**Proposition 2** *Si un code est MDS alors tout ensemble de* $k$ *coordonnées les mots codes restrincts à ces coordonnés sont distinctes (i.e. les* $k$ *composantes de tout mot code définissent le mot code).*

**Preuve** Voir preuve borne de Singleton. ■

**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 1** *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): appréciez l'élégance!

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 $\mathbb{F}_q$. A un message on associe un polynôme:

$$m = (m_0, m_2, \ldots, m_{k-1}) \leftrightarrow f_m = \sum_{i=1}^{k-1} m_i x^i.$$

Le code de Reed-Solomon (RS) est

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

On observe que pour tout message $m$ et $m'$

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

et

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

et donc (comme $deg(f_{m+m'}(X)) < k$)

$$RS(m) + RS(m') \in C$$

et

$$a \cdot RS(m) \in C.$$

Un code RS est donc linéaire. Alternativement, la linéarité se voit car l'encodage correspond à

$$(x_1, x_2, \ldots, x_n) = (m_1, m_2, \ldots, m_k) \begin{pmatrix} 1 & 1 & 1 & . & . & . & 1 \\ \alpha_1 & \alpha_2 & & . & . & . & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & . & . & . & . & . \\ . & . & . & . & . & . & \alpha_n^{k-1} \end{pmatrix}$$

avec à droite la "matrice d'évaluation" correpondant à la matrice génératrice.

Ce code a pour paramètres:

- longueur $n$
- dimension $q^k$. Pour établir ceci il suffit de montrer que tout polynôme donne un mot code diff'erent. Si il existait $f_1 \neq f_2$ t.q. $f_1(\alpha_i) = f_2(\alpha_i) \forall i$ et telles que $deg(f_1) < k$ et $deg(f_2) < k$, alors en posant
$$g = f_1 - f_2$$
on aurait que le nombres de racines de $g$ est $\geq n \geq k$ alors que $deg(g) < k$ ce qui impossible.
- une distance minimale $d = n - k + 1$. En effet
$$d = \min_{c \in C, c \neq 0} w(c)$$
et comme
$$w(c) = n - nbre\, racines$$
et que 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.*

**Observation 4** *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 $[q]$ fixé 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 5** *$RS(n, k-1) \subseteq RS(n, k)$ car les polynôme de degré $\leq k$ sont aussi de degré $\leq k-1$.*

**Observation 6** *Eliminer (ponctuer) une même coordonnée à tous les mots codes d'un code de RS(n,k) donne un code de Reed Solomon (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 $m$ (la matrice est inversible a gauche).

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 $\Lambda$ donnent les localisations des erreurs. Si l'on parvient à connaître $\Lambda$, on peut retrouver et éliminer les erreurs pour autant que leur nombre est $\leq d - 1$ (propriété MDS).

**Observation 7** *Le polynôme $\Lambda$ 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 $\Lambda$ et $f$) ce qui rend la solution possible mais complexe à trouver.

## 3.2 Relaxation du problème

**Problème 2** *Etant donné $y_1, y_2, \ldots$, trouver $\Lambda(X)$ et $f(X)$ tels que*

$$\Lambda(\alpha_i) \cdot y_i - h(\alpha_i) = 0 \quad \forall i \tag{2}$$

*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 $\Lambda$. On essaie de résoudre pour $t = 0$, $t = 1$,... jusqu'au moment où on trouve une solution pour $\Lambda$ et $h$. Si $h/\lambda$ est un polynôme de degré $< k$ alors l'algorithme produit $\hat{f} = h/\lambda$. Sinon, il déclare une erreur.

1. Comment garantir qu'une paire $(h, \lambda)$ 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

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

   puisque $d = n - k + 1$. Donc l'algorithme trouve une paire $(h, \lambda)$ pour un

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

   (la motié de la distance minimale). De plus, une de ces paires $(h, \lambda)$ est donnée par le polynôme localisateur $\lambda(X) = \prod_{e_j \neq 0}(X - \alpha_j)$, ce qui correspond donc à une solution valide.

2. Cette solution est-elle unique? Soit $(h_1, \lambda_1)$ et $(h_2, \lambda_2)$ deux solutions de (2) pour un même $t \leq \left\lceil \frac{d-1}{2} \right\rceil$. Alors

   $$h_1(\alpha_i) * \lambda_2(\alpha_i) = \lambda_1(\alpha_i) * y_i * \lambda_2(\alpha_i) = \lambda_1(\alpha_i) * h_2(\alpha_i) \qquad i = 1, 2, \ldots, n.$$

   Puisque les degrés de $h_1 * \lambda_2$ et de $h_2 * \lambda_1$ est $2t + k - 1 \leq d + k - 2 = n - 1$ et que ces polynômes sont égaux sur $n$ valeurs distinctes, ils sont égaux.

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

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

et que cette solution est correcte. De plus le décodage est de faible complexité; la résolution du système linéaire déquation plus haut peut se faire avec complexité $O(n^3)$.

# 4 Codes BCH (Bose, Ray-Chaudhuri, Hocquenghem)

Vu: pour $1 \leq k \leq n$ et $\mathbb{F}_q$ t.q. $n \leq q$ il existe un code $RS(n, k, d = n - k + 1)$.

Soit $n = q = p^m$, ou $p$ est premier et $m$ est entier. On définit le code

$$BCH_{p,m,d} \equiv RS[n, n-d+1, d]_{p^m} \cap \mathbb{F}_p^n$$

I.e., le sous-code de RS obtenu par la restriction des composantes dans le corps de base $\mathbb{F}_p$. Se décode donc comme un code RS.

Paramètres:

- longeur $n = p^m$
- distance minimale $\geq d$

Remarque:
Ces codes permettent d'atteindre la borne de Hamming pour certaines petites valeurs de $n$.

**Théorème 3**

$$\dim(BCH_{p,m,d}) \geq p^m - 1 - m \left\lceil \frac{(d-2)(d-1)}{p} \right\rceil$$

*et donc pour tout* $m, t \geq 1$ *entier* $BCH_{2,m,2t}$ *est un* $[n, n-1-(t-1)\log_2 n, 2t]_2$ *code.*

Cette classe de codes est intéressante si seulement i $t = o(n)$ (ce qui donne un taux élevé et une distance minimale faible).

# 5 Rappelons notre problème

Peut-on atteindre $\delta, R > 0$ avec une taille d'alphabet constante et une faible compléxité de codage/décodage?

- Hamming: $[n, n - \log_2(n+1), 3]_2$ d'où $R = 1 - O((\log n)/n)$ et $\delta = O(1/n)$. Taux élevé, distance faible, faible complexité.

- RS: code possible (c.f. exos) $[n = 2^t, n/2, n/2+1]_{n=2^t=q}$ d'où $R, \delta \to 1/2$ mais $q = n$...

  Idée: écrire chaque symbol sur $t$ bits. On obtient donc un code $[n \log_2 n, (n \log_2 n)/2, n/2+1]_2$. D'où $R \to 1/2$ mais $\delta \to 0$. En effet la distance minimale reste inchangée car les symboles sont "codés" avec un code de distance 1. Et si on augmentait cette distance par un véritable code correcteur d'erreurs? Ceci aboutit aux codes dit "concaténés".

# 6 Codes concaténés

Soit $q \geq 2, k \geq 1$ entiers et $Q = q^k$.

Soit

$$C_{out} : [Q]^K \to [Q]^N$$

code dit extérieur et

$$C_{in} : [q]^k \to [q]^n$$

code dit intérieur

Pour une mot

$$m = (m_1, \ldots, m_K)$$

on a

$$C_{out}(m) = [C_{out}(m)_1, \ldots, C_{out}(m)_N]$$

et ensuite on utilise $C_{in}$ et on obtient

$$[C_{in}(C_{out}(m)_1), \ldots, C_{in}(C_{out}(m)_N)] \equiv C_{in} \circ C_{out}(m)$$

le concaténation de codes $C_{in}$ et $C_{out}$.

**Théorème 4** *$C_{in} \circ C_{out}$ est un code $[n \cdot N, k \cdot K, d \cdot D]_q$. De plus, si $C_{in}$ et $C_{out}$ sont linéaires, alors $C_{in} \circ C_{out}$ est linéaire (cette deuxième partie est prouvée en exercices).*

**Corollaire 2** *Si $C_{out}$ et $C_{in}$ ont les taux $R$ et $r$ et distances $\delta_{out}$ et $\delta_{in}$, respectivement, alors $C_{in} \circ C_{out}$ a taux $R \cdot r$ et distance minimale $\delta_{out} \cdot \delta_{in}$.*

En exercice (voir borne de Zyablov) on va voir que par concaténation on peut construire des codes à faible complexité et atteignant $\delta, R > 0$ sur un alphabet de dimension donnée (qui ne grandit pas avec la longueur de bloc).