THEORIE

DE L'INFORMATION

jp.guédon

INFO 3

Théorie des codes detecteurs/correcteurs

- 1 Définitions et quelques exemples
- 2. Codes linéaires
- 2.1 Définitions
- 2.2 Codage / Décodage
- 2.3 Codes de Hamming
- 2.4 Autres codes linéaires
- 3 Codes cycliques
- 3.1 Polynome générateur
- 3.2 liens codes CRC-linéaire
- 3.3 Codes RS

Introduction

inventé par un informaticien (préhistoire 1947) désireux de mettre ses jobs le vendredi pour récupérer le résultat le lundi sur un ordinateur (style ENIAC) qui lors de chaque erreur binaire de lecture était capable de repérer cette erreur (possibilité de détection d'erreur) mais jetait le programme (pas de possibilité de correction d'erreur).

Richard Hamming (Bell Labs) inventa donc un code capable de permettre à l'ordinateur de corriger des erreurs binaires.

Introduction

L'exemple le plus simple :

Je veux transmettre soit « 0 » soit « 1 »

Cas 1

Je double le symbole et transmet 00 ou 11

Je peux recevoir un des quatre messages:

00 01 10 11

Donc, soit je reçois un mot code soit je sais qu'il y a eu une erreur : code détecteur

Introduction

Vocabulaire:

J'ai transmis des séries de longueur n=2 bits

Le nombre de vecteurs de code m est donc

2 = Card C avec C={00,11}

La dimension du code est k = log2(Card C) = 1:

c'est le nombre de bits d'information transmis pour un bloc de n éléments binaires

Introduction

L'exemple le plus simple :

Je veux transmettre soit « 0 » soit « 1 »

Cas 2

Je triple le symbole et transmet 000 ou 111

Je peux recevoir un des neuf messages:

000 001 010 100 110 101 011 111

Maintenant :

Introduction

000 001 010 100 110 101 011 111

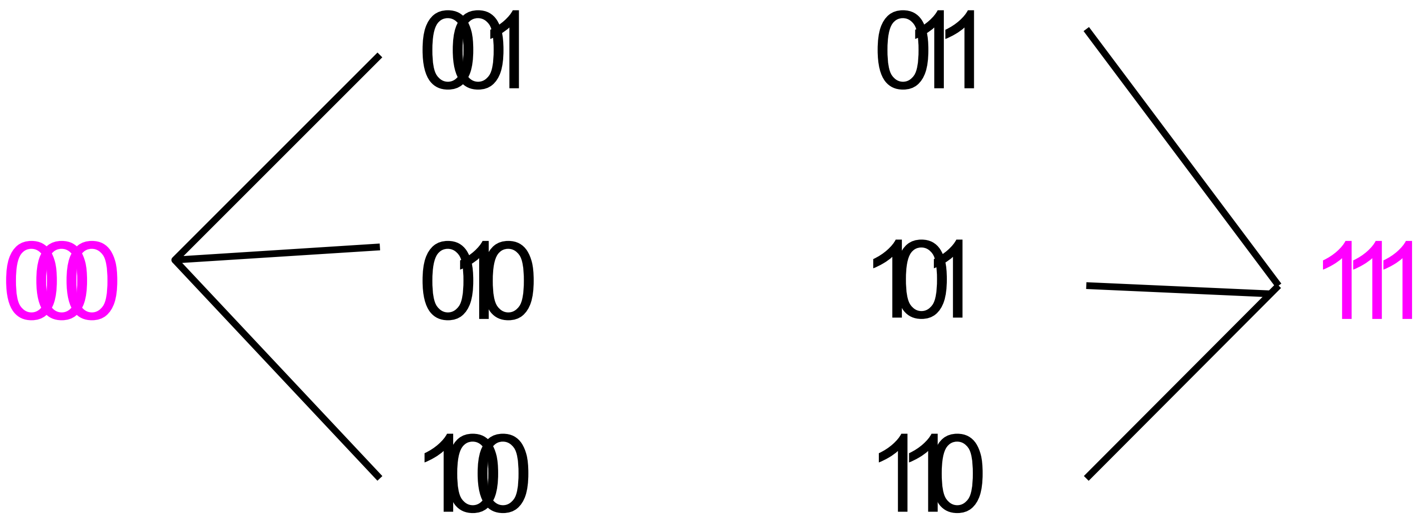
001 011

000 010 101 111

100 110

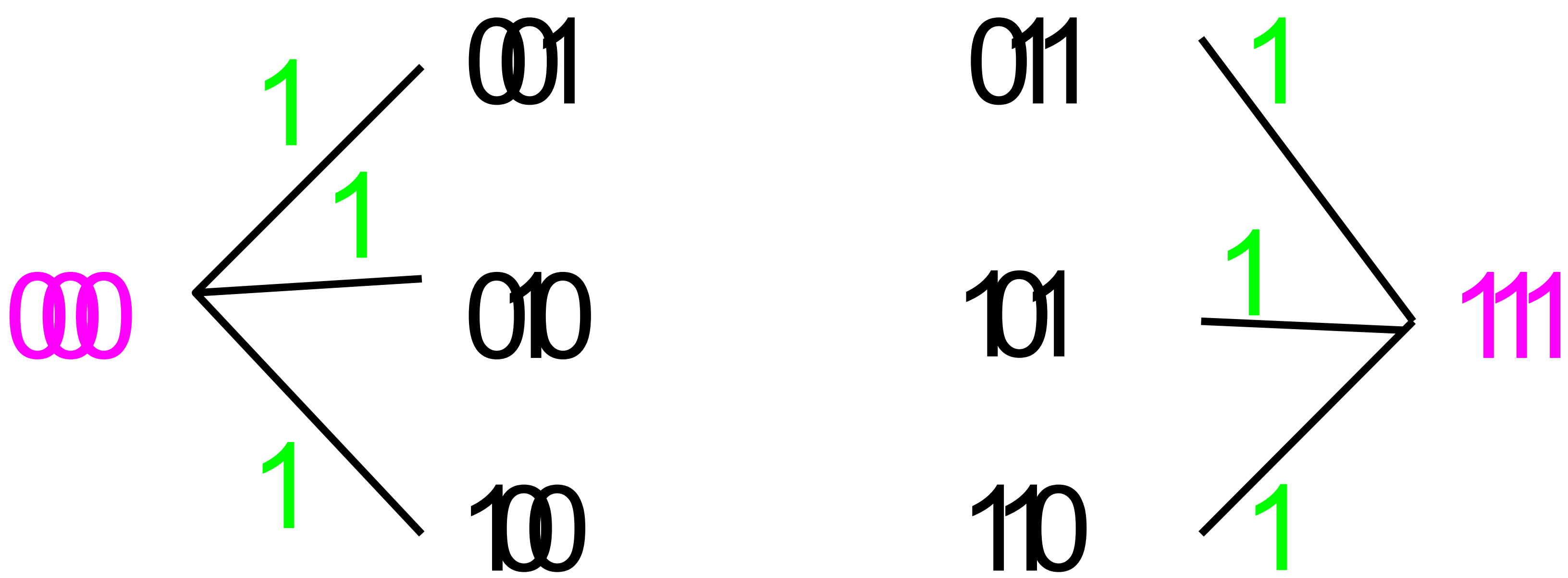
Introduction

000 001 010 100 110 101 011 111



Introduction

000 001 010 100 110 101 011 111



Codes linéaires

Paramètres d'un code

Un code C est dénoté par (n,qk,d) ou bien [n,k,d] pour un code linéaire.

La **dimension** du code est k=logq(card (C)),

la **longueur** du code est n

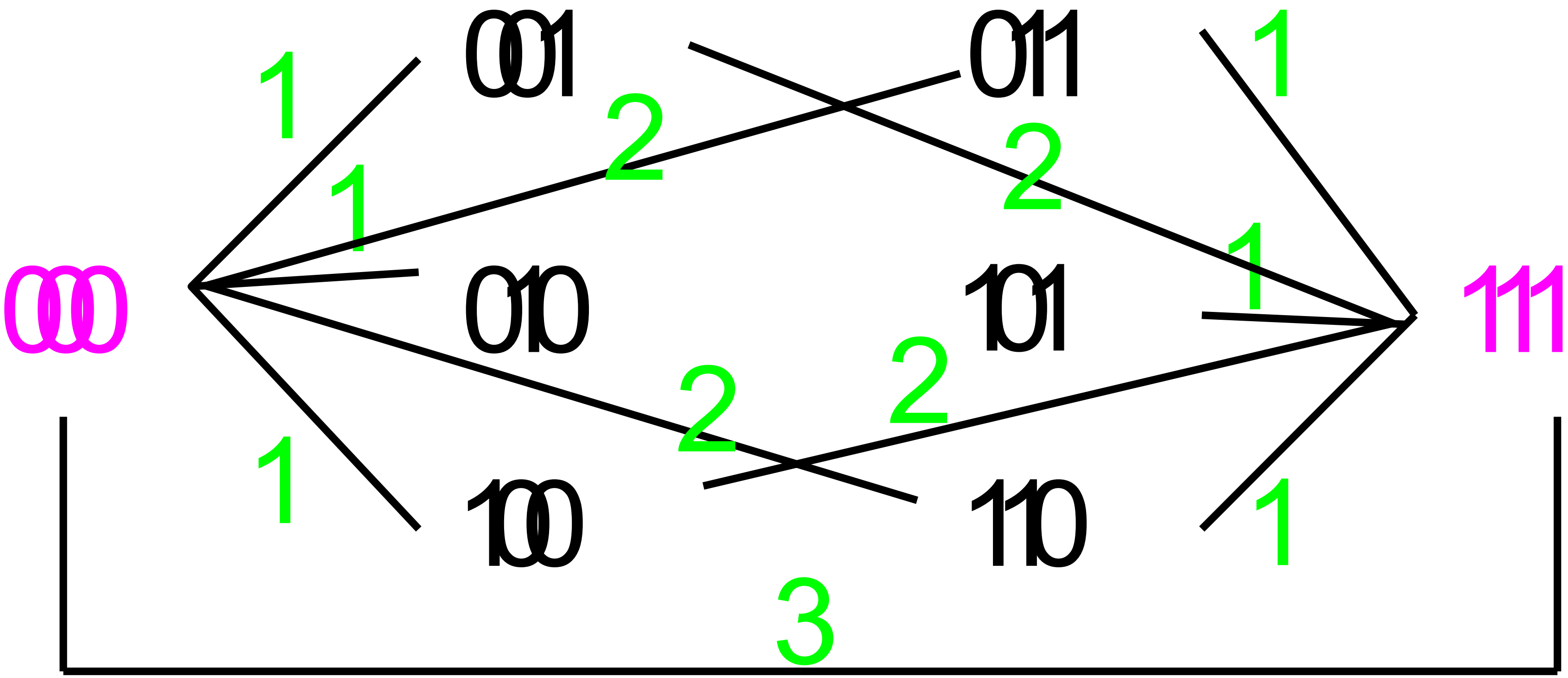
le **pouvoir de détection** du code est d

On appelle **d**(C) la plus petite distance entre deux mots code quelconques x et y de C:

$$d(C) = \min_{(x,y)\in C,\, x\neq y} d(x,y)$$

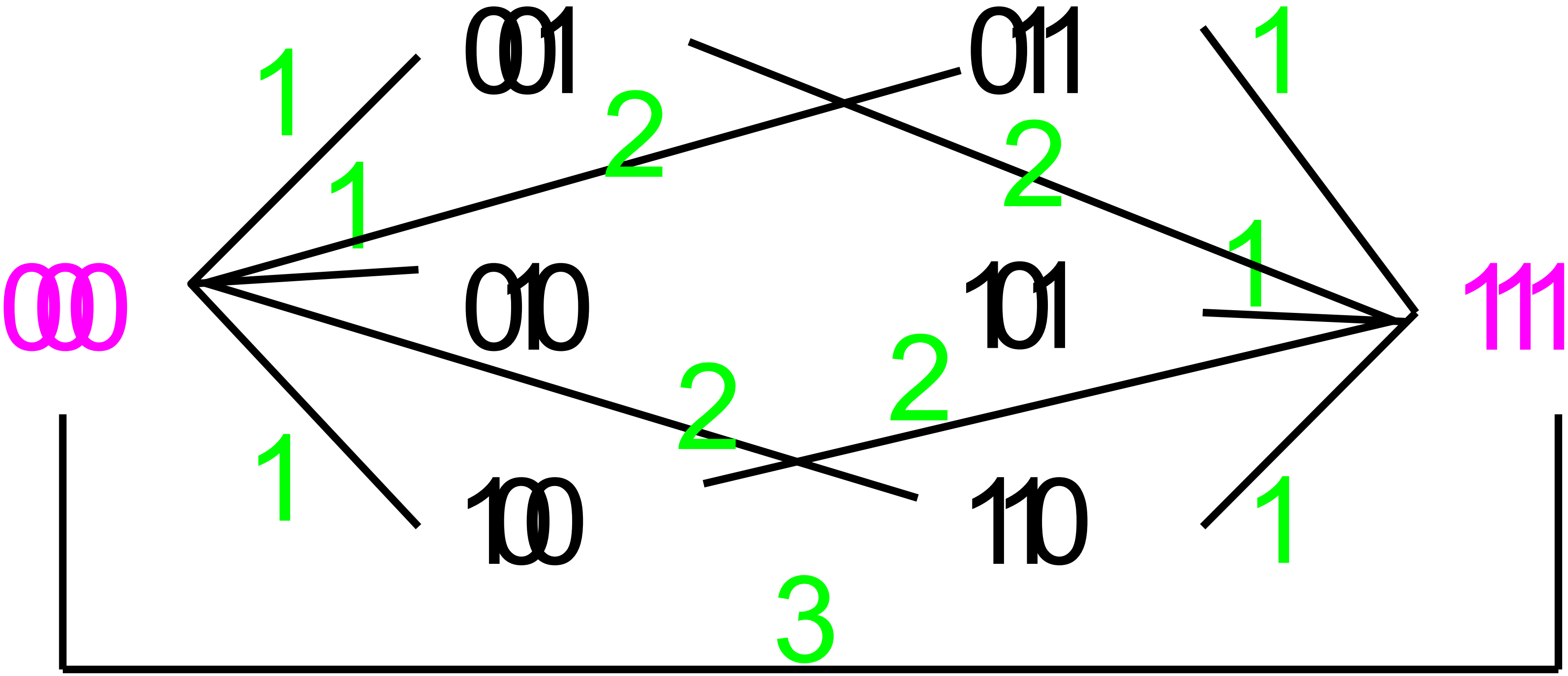
Introduction

000 001 010 100 110 101 011 111



Introduction

La distance de Hamming



Codes linéaires

Paramètres d'un code

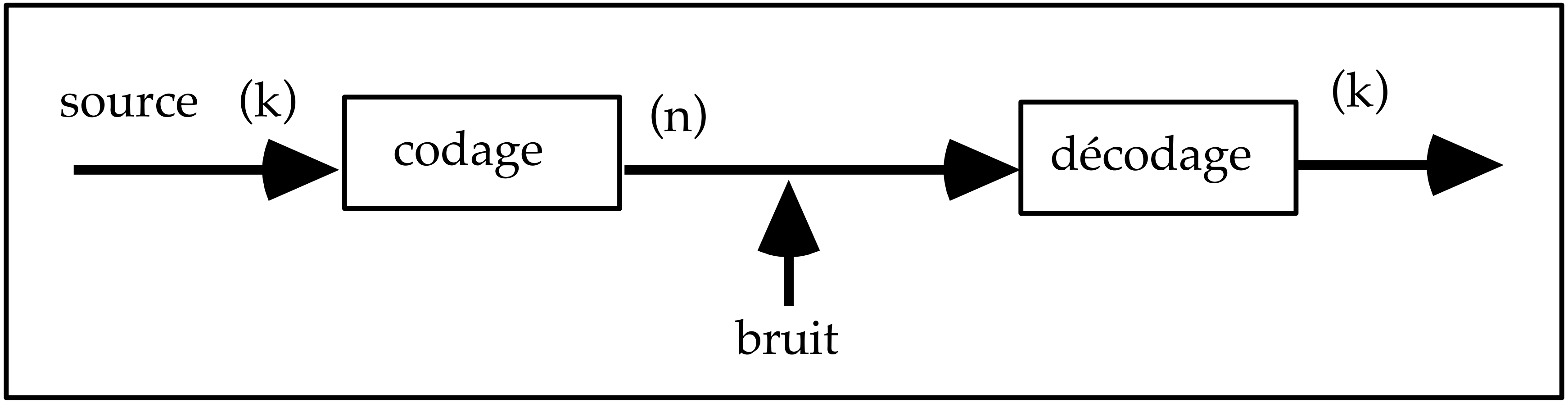
Un code est dénoté par (n,qk,d)

ou bien [n,k,d] pour un code linéaire.

La **dimension** du code est k=logq(card (C)),

la **longueur** du code est n

le **pouvoir de détection** du code est d



Paramètres d'un code détecteur

Un code est dénoté par (n,qk,d)

ou bien [n,k,d] pour un code linéaire.

La **dimension** du code est k=logq(card (C)),

la **longueur** du code est n

le **pouvoir de détection** du code est d

Pour détecter une erreur il faut moins de d erreurs

=> Si moins de d/2 erreurs le vecteur reçu est toujours plus proche de l'original que de tout autre vecteur

Paramètres d'un code correcteur

Un code est dénoté par (n,qk,d)

ou bien [n,k,d] pour un code linéaire.

La **dimension** du code est k=logq(card (C)),

la **longueur** du code est n

le **pouvoir de détection** du code est d

Pour corriger e erreurs:

un code de distance minimale d permet de corriger

$$e = floor\frac{d-1}{2}$$ erreurs

Construction de codes

situation idéale :

n petit => rapidité

M grand => efficacité en débit

d grand => efficacité en correction

Exemple : construire un code (n=? ; M=4; d=3)

Codes linéaires

Code linéaire et distance minimale d'un code

Un code linéaire C de longueur n est un sous-espace vectoriel de l'espace de dimension n : Fn = $\{0,1\}^n$

=> en fait construit sur le corps de Galois

=> c'est par combinaison linéaire que l'on va générer les (n-k) valeurs constituants le code

=> le formalisme matriciel va être adéquat pour faire cela.

Codes linéaires

Code linéaire et distance minimale d'un code

explication du mécanisme:

L'espace vectoriel est décrit à partir de k vecteurs indépendants formant une base.

Dans l'espace du code de dimension n >k, on obtient une nouvelle structure pour les n vecteurs :

ils forment une partie génératrice mais pas libre :

on parle alors de structure de FRAME.

Codes linéaires

Code linéaire et distance minimale d'un code

explication du mécanisme:

Cette structure de FRAME va permettre de montrer (cas d'un code détecteur)

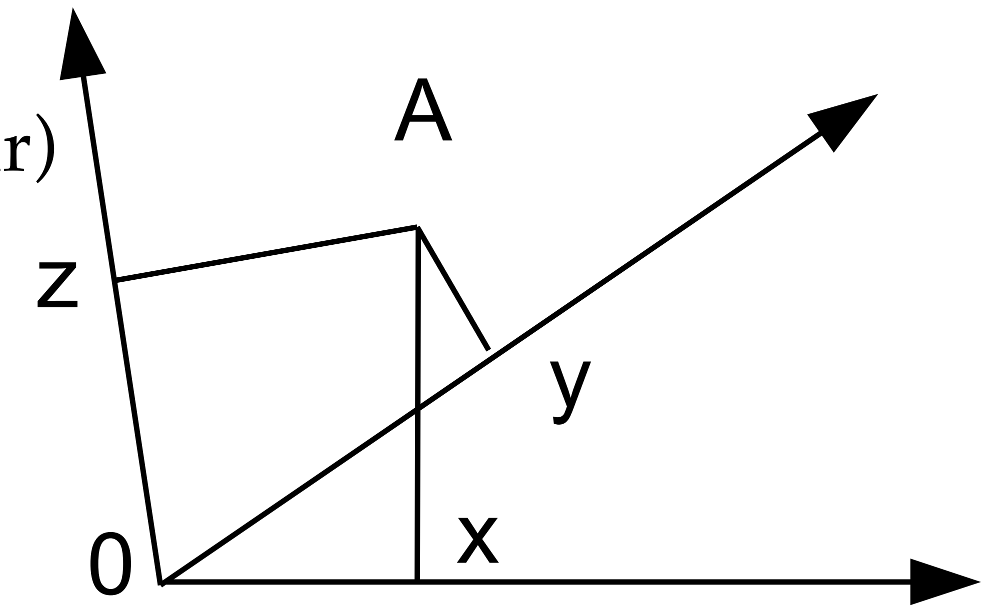
et/ou de résoudre

(cas d'un code correcteur)

les incohérences

existantes entre

ces vecteurs



Codes linéaires

Distance et poids de Hamming

La distance de Hamming d(x,y) est le nombre de composantes (parmi les n possibles) où xi ≠ yi (avec xi dans {0,1} le plus souvent). Autrement dit, dans ce cas on a:

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Codes linéaires

Distance et poids de Hamming

Le poids de Hamming w(x) est le nombre de composantes non nulles du vecteur x :

$$w(x)=\sum_{i=1}^{n}|x_i|$$

Codes linéaires

Code linéaire et distance minimale d'un code

exemple:

Pour un vecteur à transmettre de k valeurs

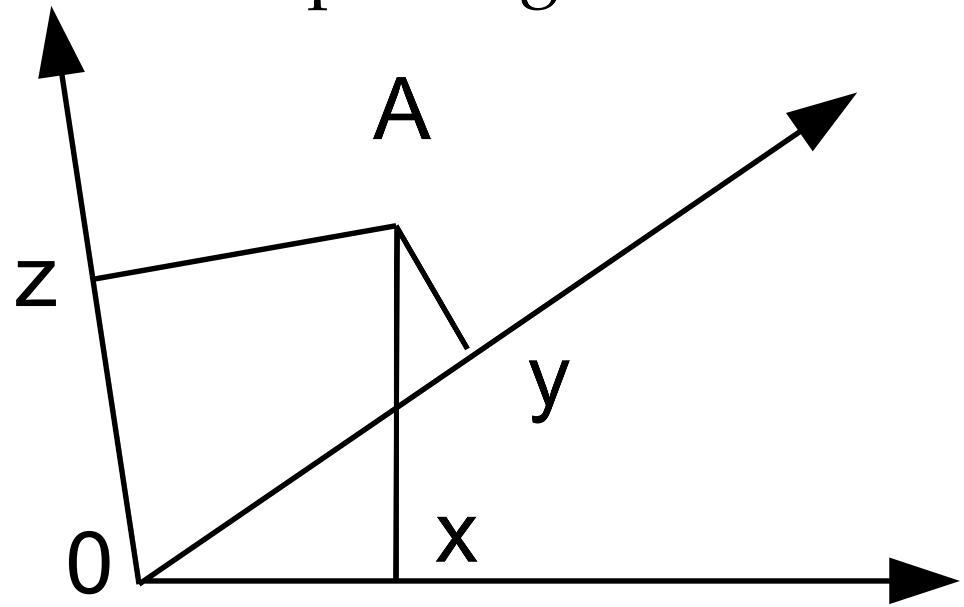
Ajoutons une valeur qui donne la parité globale

(n=k+1)

C'est le bit de parité

Ex: (0 1) => (0 1 1)

Et (1 1) => (1 1 0)



Construction de codes
à partir de plusieurs bits de parité

Soit un vecteur de 4 bits d'information u1 u2 u3 et u4

On veut calculer 3 bits de redondance

Par des équations de parité

Construction de codes
à partir de plusieurs bits de parité

Soit un vecteur de 4 bits d'information u1 u2 u3 et u4

On veut calculer 3 bits de redondance b1 b2 b3

Par des équations de parité

On met les 7 bits sur les positions suivantes:

(b1 b2 u1 b3 u2 u3 u4)

1 2 3 4 5 6 7

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité

(b1	b2	u1	b3	u2	u3	u4)
1	2	3	4	5	6	7
001	010	011	100	101	110	111

Équation de parité: 1b1 +1u1 + 1 u2 + 1u4 = 0

Construction de codes
à partir de plusieurs bits de parité

(b1	b2	u1	b3	u2	u3	u4)
1	2	3	4	5	6	7
001	010	011	100	101	110	111

Équation de parité : b2 + u1 +u3 +u4 = 0

Construction de codes à partir de plusieurs bits de parité

(b1	b2	u1	b3	u2	u3	u4)
1	2	3	4	5	6	7
001	010	011	100	101	110	111

Équation de parité : $b3 + u2 + u3 + u4 = 0$

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité

(b1 b2 u1 b3 u2 u3 u4)

1 2 3 4 5 6 7

001 010 011 100 101 110 111

b1 +u1 + u2 + u4 = 0 => u1 + u2 + u4 = b1

b2 + u1 +u3 +u4 = 0 => u1 +u3 +u4 = b2

b3 + u2 +u3 +u4 = 0 => u2 +u3 +u4 = b3

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité
exemple

(b1 b2 u1=1 b3 u2=0 u3=1 u4=0)

u1 + u2 + u4 = b1 => b1= 1

u1 +u3 +u4 = b2 => b2= 0

u2 +u3 +u4 = b3 => b3= 1

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité
exemple

(b1=1 b2=0 u1=1 b3=1 u2=0 u3=1 u4=0)

Au décodage on vérifie les équations

u1 + u2 + u4 + b1 = 0

u1 +u3 +u4 + b2 = 0

u2 +u3 +u4 + b3 = 0

Construction de codes
à partir de plusieurs bits de parité
exemple

(b1=1 b2=0 u1=1 b3=1 u2=0 u3=1 u4=0)

Au décodage on vérifie les équations

Trois équations justes = pas d'erreur

u1 + u2 + u4 + b1 = 0

u1 +u3 +u4 + b2 = 0

u2 +u3 +u4 + b3 = 0

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité
Exemple avec erreur de transmission

(b1=1 b2=0 u1=1 b3=1 u2=1 u3=1 u4=0)

Au décodage on vérifie les équations

Erreur!

u1 + u2 + u4 + b1 = 1

u1 +u3 +u4 + b2 = 0

u2 +u3 +u4 + b3 = 1

Codes linéaires

Construction de codes
à partir de plusieurs bits de parité
Exemple avec erreur de transmission

(b1=1   b2=0   u1=1   b3=1   u2=1   u3=1   u4=0)

001   010   011   100   101   110   111

Au décodage on vérifie les équations

Erreur!

u1 + u2 + u4 + b1 = 1

u1 +u3 +u4 + b2 = 0

u2 +u3 +u4 + b3 = 1

Codes de Hamming (4,7)

L'erreur donne sa position !

(b1=1 b2=0 u1=1 b3=1 u2=1 u3=1 u4=0)

001 010 011 100 101 110 111

Au décodage on vérifie les équations

Erreur!

$u1 + u2 + u4 + b1 = 1$

$u1 + u3 + u4 + b2 = 0$

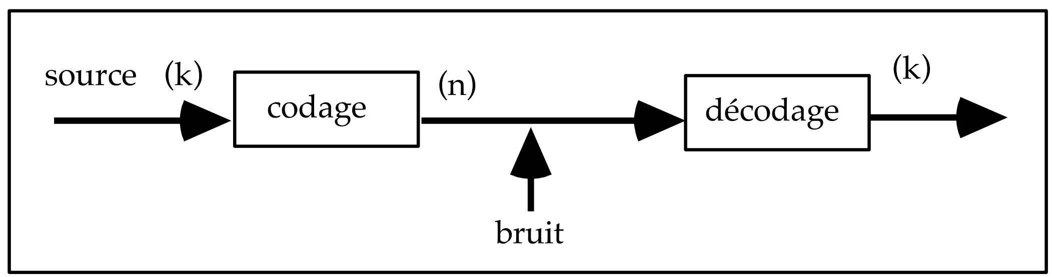
$u2 + u3 + u4 + b3 = 1$

Codes linéaires

Codage et matrice associée

Matrice génératrice G : pour un code linéaire [n,k] c'est la matrice de taille k! n dont les lignes constituent une base du code C. Le rang de la matrice est k.

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}$$



avec gij =0 ou 1 (si on opère dans Galois F2).

Codes linéaires

Codage et matrice associée

On a alors le (vecteur) mot codé **x** par le code C de matrice G à partir du (vecteur) mot **u** en faisant :

$$x^t = u^t . G$$

$$\left(x_1 x_2 \ldots x_n\right) = \left(u_1 u_2 \ldots u_k\right) . \begin{pmatrix} g_{11} & g_{12} & \ldots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}$$

[1 x n] = [1 x k] [k x n]

Codes linéaires

Codage et matrice associée

On a alors le (vecteur) mot codé **x** par le code C de matrice G à partir du (vecteur) mot **u** en faisant par transposition :

$\mathbf{x} = \mathbf{G}^t\, \mathbf{u}$

$$\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} g_{11} & g_{21} & \dots & g_{k1} \\ g_{12} & g_{22} & \dots & g_{k2} \\ \dots & \dots & \dots & \dots \\ g_{1n} & g_{2n} & \dots & g_{kn} \end{pmatrix} . \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_k \end{pmatrix}$$

Codes linéaires

Codage et matrice associée

Dans le cas ou l'on ajoutait un bit identique à celui que l'on voulait transmettre on avait k=1, n=2.

G = (g11 g12), avec gij =0 ou 1 soit tel que:

$(11)^t = (1)\, {}^t\mathbf{G}$ et $(00)^t = (0)\, {}^t\mathbf{G}$,

=> 2 équations => **G** = [1 1]

De même si l'on triple le bit on obtient **G** = [1 1 1]

Remarque : le rang de **G** est bien k = 1 dans les deux cas (dim de la base).

Codes linéaires

Codage et matrice associée

Exercice matrice de parité

On veut coder sous forme matricielle le code de parité d'un vecteur de 4 bits

(u1 u2 u3 u4 b)

Écrire la matrice **G** correspondante.

Codes linéaires

Codage et matrice associée

Exercice matrice de parité

On veut coder sous forme matricielle le code de parité d'un vecteur de 4 bits

(u1 u2 u3 u4 b) avec

u1 + u2 + u3 + u4 + b = 0

Écrire la matrice **G** correspondante.

Codes linéaires

Codage et matrice associée

Exercice matrice de parité

On veut coder sous forme matricielle le code de parité d'un vecteur de 4 bits

(u1 u2 u3 u4 b) avec

u1 + u2 + u3 + u4 + b = 0

$$G = \begin{pmatrix} 1\,0\,0\,0\,1 \\ 0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,1 \end{pmatrix}$$

Codes linéaires

Codage et matrice associée

Exercice matrice de Hamming (7,4)

On veut coder sous forme matricielle le code de Hamming (7,4)

(b1 b2 u1 b3 u2 u3 u4) avec

$u1 + u2 + u4 + b1 = 0$

$u1 + u3 + u4 + b2 = 0$

$u2 + u3 + u4 + b3 = 0$

Écrire la matrice **G** correspondante.

Codes linéaires

Codage et matrice associée

On veut coder sous forme matricielle le code de Hamming (7,4)

(b1 b2 u1 b3 u2 u3 u4) avec

u1 + u2 + u4 + b1 = 0

u1 + u3 +u4 + b2 = 0

u2 + u3 +u4 + b3 = 0 Écrire la matrice **G**

$$G^t = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{pmatrix}$$

Codes linéaires

Codage et matrice associée

Exercice

On veut coder non plus un bit mais 2 de telle sorte qu'en sortie on inverse le motif des deux bits

(ex: 01 donne 0110 ou bien 11 donne 1100)

- le bit 3 est le complément du bit 1
- le bit 4 est le complément du bit 2.
- Écrire la matrice **G** correspondante. Calculer le rang

Codes linéaires

Matrice systématique

Si l'on désire mettre tous les bit du message originel au début et mettre des bits de protection à la fin, la matrice G doit être de la forme :

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 & g_{(1k+1)} \dots & g_{1n} \\ 0 & 1 & \dots & 0 & g_{(2k+1)} \dots & g_{2n} \\ \dots & \dots & \dots & \dots & \dots \quad \dots & \dots \\ 0 & 0 & \dots & 1 & g_{(kk+1)} \dots & g_{kn} \end{pmatrix}$$

avec gij =0 ou 1

Codes linéaires

Matrice systématique

On peut écrire : $x^t = u^t \, G = [u^t \; ! \quad v^t]$.

On écrit donc la matrice $G = [I_{k,k} \quad ! \quad P_{k,n-k}]$.

L'avantage majeur est bien sûr de ne pas avoir a modifier le message à envoyer ,

on rajoute juste les bits de code à la fin.

Codes linéaires

Code dual C⊥

Rappel :

2 vecteurs **a** et **b** sont orthogonaux

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

Dans les codes linéaires, le produit de 2 vecteurs correspond à la forme bilinéaire symétrique suivante: $\mathbf{a}^t\mathbf{b} = \sum_i a_i b_i$

Codes linéaires

Code dual C⊥

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

Les vecteurs **w** de Fn orthogonaux aux vecteurs **x** du code C[n,k] forment un sous-espace vectoriel dual de celui engendré par C.

On peut en déduire un code dual de C du type

C⊥ [n,n—k] et de matrice génératrice H tel que

$$\mathbf{GH}^t = \mathbf{0}$$

Codes linéaires

exemple

G [k=3,n=5] H [n=5,n—k=2] telle que

GH^t = **0**

$$G=\begin{pmatrix}1\,0\,0\,1\,1\\0\,1\,0\,1\,0\\1\,0\,1\,1\,1\end{pmatrix}$$

$$\begin{pmatrix}1\,0\,0\;\;1\,1\\0\,1\,0\,1\,0\\0\;\;0\,1\,1\,1\end{pmatrix} \text{x} \begin{pmatrix}1\,1\\1\,0\\1\,1\\1\,0\\0\,1\end{pmatrix} = \begin{pmatrix}0\,0\\0\,0\\0\,0\end{pmatrix}$$

$$H^t=\begin{pmatrix}1\,1\\1\,0\\1\,1\\1\,0\\0\,1\end{pmatrix} \quad G.H^t=\begin{pmatrix}0\,0\\0\,0\\0\,0\end{pmatrix}$$

[3,5] x [5,2] = [3, 2]

Codes linéaires

Code dual C⊥

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

La dimension de l'espace dual est (n — k).

Pour un code de matrice $\mathbf{G} = [\mathbf{I}_{k,k}\ \partial \mathbf{P}_{k,n-k}]$.

le code dual C⊥ [n,n—k] est défini par :

$\mathbf{H} = [\mathbf{—P}^t{}_{k,n-k}\ \partial\ \mathbf{I}_{n-k,n-k}]$.

Codes linéaires

Code dual C⊥

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

démo

$$\mathbf{GH}^t = [\mathbf{I}_{k,k} \partial \mathbf{P}_{k,n-k}][-\mathbf{P}^t{}_{k,n-k} \partial \mathbf{I}_{n-k,n-k}]^t$$

donne

$$\mathbf{GH}^t = -\mathbf{P} + P = 0$$

Remarque: si on ne considère que alphabet binaire: $-P = \mathbf{P}$ (car $-1 = 1$).

Codes linéaires

Code dual C⊥

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

La matrice $\mathbf{H}^t$ est appelée matrice de parité du code C[n,k].

Si c est un mot de C alors on $\mathbf{cH}^t = \mathbf{0}$.

- Cela caractérise les mots code c

- (les autres n'en sont pas)

(La démo est simple : C = (C⊥)⊥).

Codes linéaires

Code dual C⊥

Si c est un mot de C alors on $\mathbf{c}\mathbf{H}^t = \mathbf{0}$.

=> Algorithme de décodage d'un code linéaire :

Je reçois le vecteur c

Je calcule le syndrome s par : $\mathbf{c}\mathbf{H}^t = \mathbf{s}$

Si le syndrome s est nul pas d'erreur détectée

Sinon erreur détectée

Codes linéaires

Code dual C⊥

$(a \perp b) \Leftrightarrow \mathbf{a}^t\mathbf{b} = 0.$

Attention il existe en général pour les codes des vecteurs isotropes c'est à dire des vecteurs codes $a \neq 0$

tels que $\mathbf{a}^t\mathbf{a} = \Sigma\ ai.ai = 0$

parce que dans F2 : $1 + 1 = 0$

Exemple: a=(1 0 0 0 1)

Algorithme décodage et correction

Début

Recevoir b

Former $s=bH^t$

Si s = 0 Alors pas d'erreur détectée

Sinon â <= b!   e

Fin

Codes linéaires

Décodage et correction

Soit

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Déterminer G et H

Codes linéaires

Décodage et correction

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Donne

$$G = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$H = \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

Codes linéaires

Décodage et correction

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Soit b le mot reçu : on forme s=bHt:

$$H = \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

Exprimez les valeurs de s dans une table

Trouvez pour chacun la valeur de e telle que â= b+e

Décodage et correction

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Soit b le mot reçu : on forme $s=bH^t = (s1\ s2\ s3)$

$$H = \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

Si $s=(0\ 0\ 0)$ alors $e=(0\ 0\ 0\ 0\ 0\ 0)$

Si $s=(0\ 0\ 1)$ alors

$b1+b2+b3+b4=0$ et $b1+b3+b5=0$ et $b2+b3+b6=1$

Codes linéaires

Décodage et correction

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Soit b le mot reçu : on forme $s=bH^t = (s1\ s2\ s3)$

Si $s=(0\ 0\ 0)$ alors $e=(0\ 0\ 0\ 0\ 0\ 0)$

Si $s=(0\ 0\ 1)$ alors

$b1+b2+b3+b4=0$ donne $y+z$ faux

$b1+b3+b5=0$ => changer le dernier

$b2+b3+b6=1$ bit : $e= (000\ 001)$

Codes linéaires

Décodage et correction

Le vecteur $a=(x\ y\ z)G^t = (x\ y\ z\ x+y+z\ x+z\ y+z)$

Soit b le mot reçu : on forme $s=bH^t = (s1\ s2\ s3)$

Si

Si s =(0 1 1) alors	donne y+z et x+z faux
b1+b2+b3+b4=0	mais x+y+z vrai
b1+b3+b5=1 =>	changer x et y
b2+b3+b6=1	2 bits soit : e= (110 000)

Décodage et correction

Le vecteur a=(x y z)G^t = (x y z x+y+z x+z y+z)

Soit b le mot reçu : on forme s=bH^t et â= b!   e

Finalement on obtient (table des distances):

s	000	001	010	011
e	000000	000001	000010	110000
s	100	101	110	111

Codes linéaires

Décodage et correction

pour le code de Hamming (7,4) On a

$$G = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{pmatrix}$$

H = ??? (on ne sait pas faire car pas systématique!)

Décodage et correction

pour le code de Hamming (7,4) On a

$$G = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{pmatrix}$$

et

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

Codes linéaires

Décodage et correction

pour le code de Hamming (7,4) On a

$$\mathbf{G} = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0\,0 \\ 0\,1\,0\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{pmatrix} \qquad \mathbf{H} = \begin{pmatrix} 0\,0\,0\,1\,1\,1\,1 \\ 0\,1\,1\,0\,0\,1\,1 \\ 1\,0\,1\,0\,1\,0\,1 \end{pmatrix}$$

Preuve : $\mathbf{GH}^t = \mathbf{0}$

Codes linéaires

Décodage et correction

pour le code de Hamming (7,4) On a

$$\mathbf{GH^t} = \begin{pmatrix} 1&1&1&0&0&0&0 \\ 1&0&0&1&1&0&0 \\ 0&1&0&1&0&1&0 \\ 1&1&0&1&0&0&1 \end{pmatrix} \times \begin{pmatrix} 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&1&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{pmatrix} = \begin{pmatrix} 0&0&0 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{pmatrix}$$

$$[4\ 7] \quad \times \quad [7\ 3] = [4\ 3]$$

Décodage et correction

pour le code de Hamming (7,4) on a

Soit b le mot reçu : on forme $s=bH^t$

s	000	001	010	011
Num bit erreur	0	1	2	3
S	100	101	110	111
Num bit erreur	4	5	6	7

Codes cycliques

introduction

La construction et l'utilisation de ces codes sont très simples ce qui expliquent leur renommée.

Ils sont bien adaptés à la détection d'erreurs indépendantes et par paquets.

Ils contiennent les classes de codes de Hamming,

de Reed-Muller, les BCH (Bose-Chaudhuri-Hocquenghem) etc.

Codes cycliques

introduction

étude sur le Corps de Galois à 2 éléments (+1 = -1)

ca marche aussi pour les corps de Galois à q éléments (mais plus complexe)

Les vecteur mots de n bits **a** = (a0 a1 a2 ... an—1) sont associés aux coefficients du polynôme

$$A(x)=a_0+a_1 x^1+a_2 x^2+\ldots+a_{n-1} x^{n-1}$$

Codes cycliques

Polynome générateur

Un code C est cyclique si l'ensemble de ses mots est invariant par décalage circulaire.

{**a** = (a0 a1 a2 ... an—1)} ∈ C

⇔ {**a1** = (an—1 a0 a1 a2 ... an—2) } ∈ C

L'identification avec les polynômes donne

$$A(x) \in C <=> \{ x.A(x) - a_{n-1}(x^n - 1) \in C \}$$

$$A(x) = a_0 + a_1 x^1 + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

passage Codes cycliques - linéaires

Rappel :

2 vecteurs **a** et **b** sont orthogonaux ⇔ $\mathbf{a}^t\mathbf{b} = 0$.

Dans les codes cycliques, le produit de 2 vecteurs correspond à la forme bilinéaire symétrique suivante:

$$a^t b = \sum_{i=1}^{n} a_i b_{n-1-i}$$

Codes cycliques

Polynome générateur

Soit

M la trame de longueur k que l'on désire émettre.

F le calcul de n bits de redondance générés

T la concaténation de (M | F) de (k+n) bits

P le polynôme utilisé (n+1) bits car de degré n

On a : $T = M\,x^n + F$.

Codes cycliques

Polynome générateur

On a : $T = M\,xn + F$.

Pour décoder :

Si on divise $M\,xn$ par P on a :

$M\,xn \;/\; P = Q + R / P$

On jette le quotient Q et on garde R. On prend $F = R$ et c'est tout.

Codes cycliques

Polynome générateur

On a : T = M xn + F.

Pour décoder :

Si on divise M xn par P on a : M xn / P = Q + R/P

On prend F = R et c'est tout.

bonne idée car T est alors divisible par P :

démo: T/P = (Mxn+R)/P= Q + R/P +R/P

mais R + R = 0 (on opère dans F2) donc T/P = Q

Codes cycliques

Exemple codage

P = (1 0 1) et M = (1 0 0 1)

$$P(x) = x^2 - 1$$

$$M(x) = x^3 - 1$$

$$\frac{x^n M(x)}{P(x)} = \frac{x^5 - x^2}{x^2 - 1}$$

On fait la division dans GC(2)

$$\frac{x^5 - x^2}{x^2 - 1} = (x^3 + x + 1) + \frac{(x+1)}{x^2 - 1}$$

On trouve :

$$Q(x) = x^3 + x + 1$$

$$R(x) = x + 1$$

Donc T = (M I R) = (1 0 0 1 1 1)

Codes cycliques

Exemple décodage

On recoit T = (1 0 0 1 1 1)

$$P(x) = x^2 - 1$$

On prend M = (1 0 0 1)

$$M(x) = x^3 - 1$$

On fait la division dans GC(2) :

$$\frac{x^n M(x)}{P(x)} = \frac{x^5 - x^2}{x^2 - 1}$$

On trouve : $R(x) = x + 1$

Donc on valide le CRC reçu

exemples de Codes cycliques

Les codes CRC les plus utilisés

CRC-12 : $x12 + x\ 11 + x\ 3 + x2 + \ +1$.

Utilisé pour des caractères codés sur 6 bits (génère trame de 12 bits).

CRC-16 : $x16 + x15 + x2 + 1$

CRC-CCITT : $x16 + x12 + x5 + 1$

Utilisé pour transmission de caractères sur 8 bit

CRC-32 : $x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$

IEEE-802.x et DoD

passage Codes cycliques - linéaires

Polynome générateur

DEF Un polynome générateur d'un code cyclique C est le polynôme non nul normalisé de plus bas degré de C

Un mot d'un code cyclique est un multiple de P(x)

Un polynome divisant $\left(x^n - 1\right)$ définit un CRC

passage Codes cycliques - linéaires

Polynome générateur

Exemple sur CG(2) : n=7

On a $$\left(x^7-1\right)=\left(1+x\right)\left(1+x+x^3\right)\left(1+x^2+x^3\right)$$

Soit $$g\left(x\right)=\left(1+x+x^3\right)$$

Donc $$h\left(x\right)=\frac{\left(x^7-1\right)}{g\left(x\right)}=\left(1+x\right)\left(1+x^2+x^3\right)$$

soit $$h\left(x\right)=\left(1+x+x^2+x^4\right)$$

Le code est de dimension k = n – m = 7 – 3 = 4

passage Codes cycliques - linéaires

Polynome générateur

Un polynome divisant $\left(x^n - 1\right)$ définit un CRC

Soit $h(x) = (x^n - 1) / g(x)$

Le polynome h est générateur du code dual

g(x) est de degré m

Le code est de dimension k = n – m

Un élément {C(x)} appartient Code <=> g(x) divise C(x)

Polynôme générateur

$$1.g(x) = 1x^0 + 1x^1 + 0x^2 + 1x^3$$

$$x.g(x) = 0x^0 + 1x^1 + 1x^2 + 0x^3 + 1x^4$$

$$x^2.g(x) = 0x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 1x^5$$

$$x^3.g(x) = 0x^0 + 0x^1 + 0x^2 + 1x^3 + 1x^4 + 0x^5 + 1x^6$$

Polynome générateur

$$G = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1\,0\,1 \end{pmatrix}$$

$$\left(1.g(x) = 1\mathrm{x}^0 + 1\mathrm{x}^1 + 0\mathrm{x}^2 + 1\mathrm{x}^3\right)$$

$$\left(x.g(x) = 0\mathrm{x}^0 + 1\mathrm{x}^1 + 1\mathrm{x}^2 + 0\mathrm{x}^3 + 1\mathrm{x}^4\right)$$

$$\left(x^2.g(x) = 0\mathrm{x}^0 + 0\mathrm{x}^1 + 1\mathrm{x}^2 + 1\mathrm{x}^3 + 0\mathrm{x}^4 + 1\mathrm{x}^5\right)$$

$$\left(x^3.g(x) = 0\mathrm{x}^2 + 1\mathrm{x}^3 + 1\mathrm{x}^4 + 0\mathrm{x}^5 + 1\mathrm{x}^6\right)$$

passage Codes cycliques - linéaires

Polynome générateur

Exemple sur CG(2)

On avait $(x^7 - 1) = (1 + x)(1 + x + x^3)(1 + x^2 + x^3)$

Soit $\boxed{g(x) = (1 + x + x^3)}$

Donc h(x) = (x7 – 1) / g(x) = (1+x)(1+x2+x3)

cad $\boxed{h(x) = (1 + x + x^2 + x^4)}$

Le code est de dimension k = n – m = 7 – 3 = 4

Polynome dual

x2.h(x) = 1x6 + 0 x5 + 1x4 + 0x3 + 1 x2 + 0x +0

x.h(x) = 0x6 + 1 x5 + 0x4 + 1x3 + 1 x2 + 1x +0

1.h(x) = 0x6 + 0 x5 + 1x4 + 0x3 + 1 x2 + 1x +1

Polynome dual

$x^2.h(x) = 1x^6 + 0\,x^5 + 1x^4 + 0x^3 + 1\,x^2 + 0x + 0$

$x.h(x) = 0x^6 + 1\,x^5 + 0x^4 + 1x^3 + 1\,x^2 + 1x + 0$

$1.h(x) = 0x^6 + 0\,x^5 + 1x^4 + 0x^3 + 1\,x^2 + 1x + 1$

Polynome et matrice duale

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{array}{l} \left(x^{2.h}(x) = 1x^6 + 0x^5 + 1x^4 + 1x^3 + 1x^2 + 0x + 0\right. \\ \left(x.h(x) = 0x^6 + 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x + 0\right. \\ \left(1.h(x) = 0x^6 + 0x^5 + 1x^4 + 0x^3 + 1x^2\right) \end{array}$$

et

$$G.H^t = 0$$

DU CODE CYCLIQUE VERS …

$G.H^t = 0$

En réalité on retrouve Hamming (7,4)

Codes MDS

Code MDS Maximum Distance Separable :

Codes « parfaits » :

On utilise entièrement la redondance produite

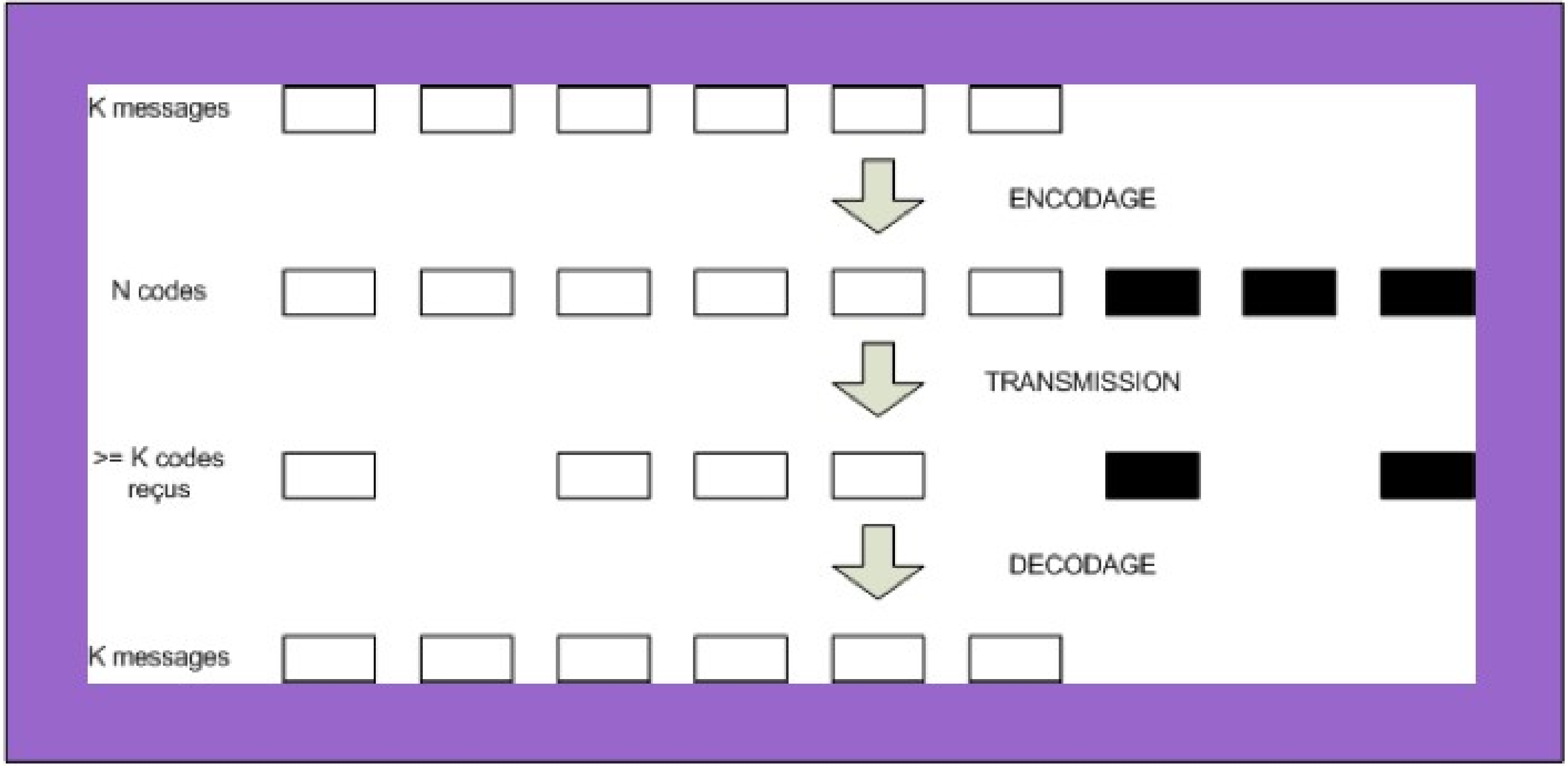
Exemple de codes

Code de Golay binaire

RS251-255

Codes MDS

Code Reed Solomon RS k-n



Codes MDS

wikipedia codes MDS Reed Solomon : RS (k-n)

Imaginons un bloc de 3 nombres que l'on souhaite transmettre :
02 09 12

Ajoutons deux nombres de redondance d'information. Le premier est la somme des 3 nombres : 02 + 09 + 12 = 23

Le second est la somme pondérée des 3 nombres, chacun est multiplié par son rang : 02×1 + 09×2 + 12×3 = 56

À la sortie du codeur, le bloc à transmettre est :

02 09 12 23 56

Codes MDS

wikipedia codes MDS Reed Solomon : RS (k-n)

Suite à perturbation, le récepteur reçoit : 02 13 12 23 56

À partir des données reçues, le décodeur calcule :

Sa somme simple : 02 + 13 + 12 = 27

Sa somme pondérée : 02×1 + 13×2 + 12×3 = 64

La différence entre la somme simple calculée (**27**) et celle reçue (23) indique la valeur de l'erreur : 4 (**27**-23 = 4)

La différence entre la somme pondérée calculée (**64**) et celle reçue (56), elle-même divisée par la valeur de l'erreur indique la position où l'erreur se trouve : 2 ((**64**-56) / 4 = 2). Il faut donc retirer 4 au nombre du rang 2.

Le bloc original est donc 02 (13-4=09) 12 23 56

Codes MDS

wikipedia codes MDS Reed Solomon : RS (k-n)

Lors d'une transmission sans perturbation, les différences des sommes simples et des sommes pondérées sont nulles.

La longueur maximale d'un code de Reed–Solomon est définie comme : $n = k + 2t; \;\; n = 2m - 1$

Avec :

m : nombre de bits par symbole ;
k : nombre de symboles d'information, appelé charge utile ;
n: nombre de symboles transmis (charge utile et correction d'erreur) ;
$2t$: nombre de symboles de contrôle.

Si la localisation des erreurs n'est pas connue à l'avance — ce qui est le cas en pratique — le codage Reed-Solomon sait corriger t=$(n - k) / 2$ erreurs.

Codes MDS

wikipedia codes MDS Reed Solomon : RS (k-n)

Transmission par satellite

Pour le DVB, le codage est RS (204, 188, t=8)

Transmission de données

En ADSL ,

le codage est souvent RS (240,224,t=8) ou encore RS (255,239,t=8).

Codes MDS

Code de Golay binaire

(utilisé dans Voyager années 80)

Mot code longueur n, k infos

Code binaire de longueur n contient au plus 2^n mots code mais 2^k messages différents

Soit t le nombre d'erreurs que le code (n,k) corrige

Code parfait : chaque mot est à distance de t d'un seul mot code

Codes MDS

code de Golay parfaits C(23,12) t=7 et q=2

$$C_{23}^{0} + C_{23}^{1} + C_{23}^{2} + C_{23}^{3} = 2^{11} = 2^{23-12}$$

On a $x^{23} - 1 = (1+x)g1(x)g2(x)$

Avec

$g1(x)= 1 +x^2+x^4+x^5+x^6+x^{10}+x^{11}$

$g2(x)=1 +x+x^5+x^6+x^7+x^9+x^{11}$

Codes MDS

Code de Golay binaire

Code parfait : chaque mot est à distance de $t=2$ d'1 seul mot code

<=> $d_{min} = 2t+1=5$

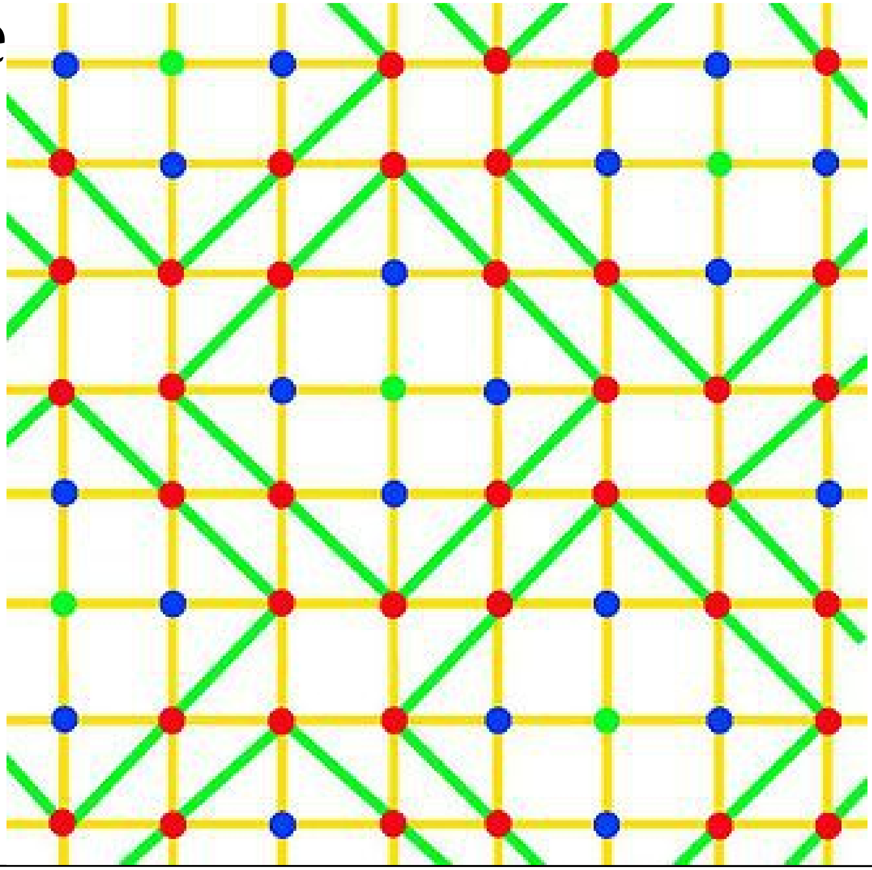
Point du code : vert

Distance entre vert =5=dmin

Point bleu distance de 1

Point rouge distance de 2

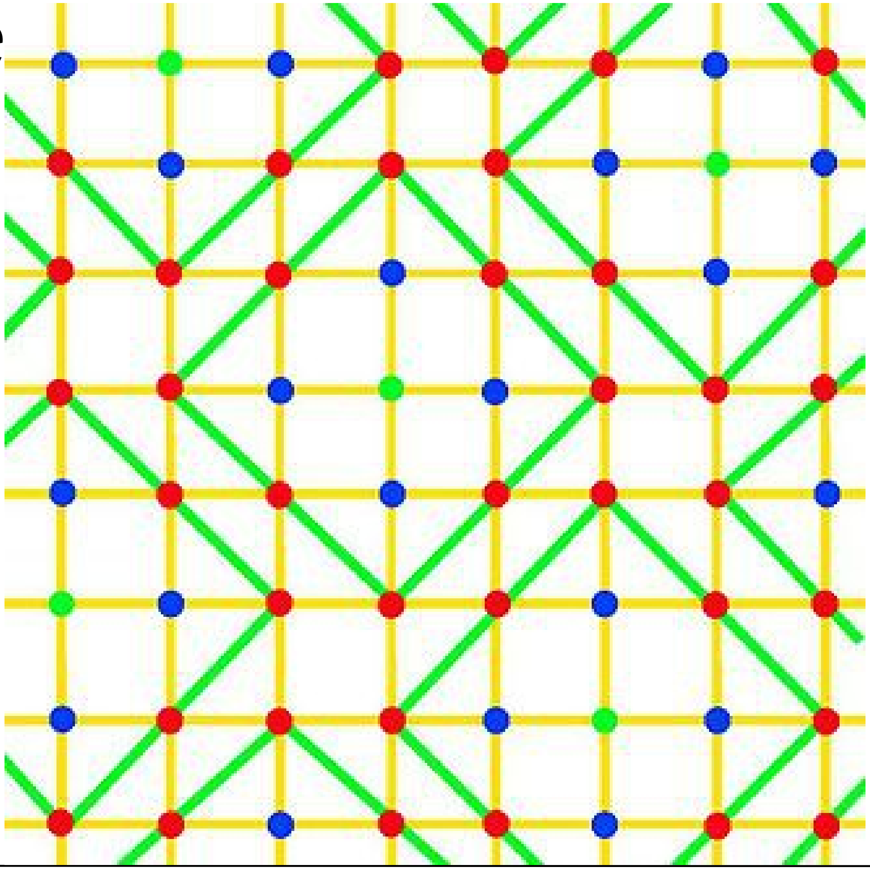
Pas d'autre point = partition de l'espace



Codes MDS

Code de Golay binaire

Code parfait : chaque mot est à distance de t=2 d'1 seul mot code



<=> dmin = 2t+1=5

Point du code : vert

Distance entre vert =5=dmin

Point bleu distance de 1

Point rouge distance de 2

Pas d'autre point = partition de l'espace

Codes MDS

Borne de Hamming

SI code (n,k) avec alphabet de q éléments et

dmin = 2t+1

ALORS

$$\left(q^n \geq q^k \sum_{(i=0)}^{(t)} Cn^i (q-1)^i\right)$$

Pour q=2 :

$$\left(2^n \geq 2^k \sum_{(i=0)}^{t} Cn^i\right)$$

Codes MDS

Code parfait <=> atteint la Borne de Hamming

Il existe 2 codes de Golay parfaits :

C(11,6) t=5 mais q=3

Et

C(23,12) t=7 et q=2

Codes MDS

Code parfait <=> atteint la Borne de Hamming

Il existe 2 codes de Golay parfaits :

C(11,6) t=5 mais q=3 Et C(23,12) t=7 et q=2

Seuls 2 autres codes sont parfaits:

Hamming (2^m -1-m) avec dmin=3

code répétition à 2 mots code vu en intro du cours

Codes MDS

code de Golay parfaits C(23,12) t=7 et q=2

Golay a remarqué :

$$C_{23}^{0} + C_{23}^{1} + C_{23}^{2} + C_{23}^{3} = 2^{11} = 2^{23-12}$$

Ce qui indique qu'il peut existe un code parfait (23,12) qui corrige jusqu'à 3 erreurs binaires

et détecte 4

En 1949, Marcel Golay, matheux Suisse, l'a trouvé et c'est le seul pour cette dimension

Codes MDS

code de Golay parfaits C(23,12) t=7 et q=2

On peut encore ajouter un bit de parité sur le code

Parfait et avoir un code étendu de Golay (24,12)

Le code obtenu est semi-parfait mais t=8 et le taux de codage donne juste un débit R=1/2 (=12/24)

Codes MDS

code de Golay parfaits C(11,6) t=2 et q=3

Golay a remarqué :

$$C_{11}^{0} + 2\,C_{11}^{1} + 4\,C_{11}^{2} = 243 = 3^5$$

$$1 + 2.11 + 4.\,(220/4) = 243$$

Ce qui indique qu'il peut exister un code parfait de 3^6 spheres de rayon t=2

dmin=5 donc corrige 2 erreurs

Avec un bit de parité Golay étendu C(12,6) et R=1/2

Théorie de l’ Information

- **merci**