

Théorie de l'Information, codage correcteur d'erreurs

Gilles Zémor

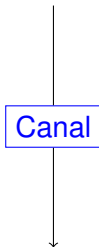
Institut de Mathématiques de Bordeaux

Novembre 2008, ENSTA

Canal binaire symétrique

Émission : vecteur de n bits

0 1 1 0 1 0 ----- 1 1



Canal binaire symétrique

Émission : vecteur de n bits

0 1 1 0 1 0 ----- 1 1

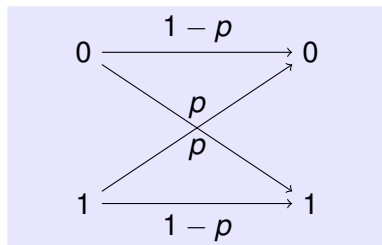
Canal

erreurs aléatoires

0 0 1 1 1 0 ----- 1 1

Canal binaire symétrique

Chaque symbole (bit) subit indépendamment :



(Vecteur reçu) = (Vecteur émis) + (Vecteur erreur)

$$\begin{aligned}\#erreurs &= \text{Poids}(\text{vect erreur}) \\ &= \text{distance de Hamming}(\text{émis, reçu}) \\ &\approx pn\end{aligned}$$

distance de Hamming(\mathbf{x}, \mathbf{y}) = $d_H(\mathbf{x}, \mathbf{y})$ = poids($\mathbf{x} - \mathbf{y}$).

Théorème de Shannon

La **capacité** du canal binaire symétrique est $C = 1 - h(p)$.

Théorème.

Pour tout $\varepsilon > 0$, il existe un **code** $C \subset \{0, 1\}^n$, de rendement $R \geq 1 - h(p) - \varepsilon$, i.e. de cardinalité $|C| \geq 2^{n(1-h(p)-\varepsilon)}$, tel qu'avec probabilité arbitrairement proche de 1, le mot de code le plus proche (pour la distance de Hamming) du vecteur reçu est le mot de code émis.

$C = 1 - h(p)$: le sup des rendements avec lesquels on peut communiquer de manière fiable.

Théorème de Shannon

La **capacité** du canal binaire symétrique est $C = 1 - h(p)$.

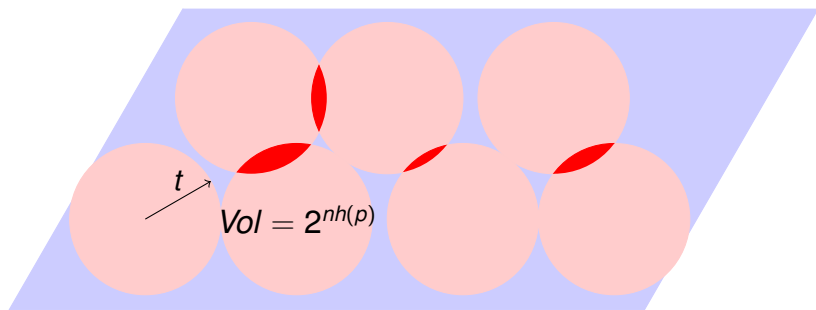
Théorème.

Pour tout $\varepsilon > 0$, il existe un **code** $C \subset \{0, 1\}^n$, de rendement $R \geq 1 - h(p) - \varepsilon$, i.e. de cardinalité $|C| \geq 2^{n(1-h(p)-\varepsilon)}$, tel qu'avec probabilité arbitrairement proche de 1, le mot de code le plus proche (pour la distance de Hamming) du vecteur reçu est le mot de code émis.

$C = 1 - h(p)$: le sup des rendements avec lesquels on peut communiquer de manière fiable.

Preuve. Il suffit de choisir le code C au hasard !

Interprétation géométrique du théorème de Shannon



Volume d'une boule de Hamming de rayon pn :

$$1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{pn} \approx 2^{nh(p)}.$$

Les boules centrées en les mots de C aléatoire, $|C| = 2^{n(1-h(p)-\varepsilon)}$, couvrent presque l'espace $\{0, 1\}^n$ et sont presque disjointes.

Codes linéaires

une matrice \mathbf{H} de dimension $r \times n$ définit la fonction **syndrome** :

$$\begin{aligned}\sigma : \{0, 1\}^n &\rightarrow \{0, 1\}^r \\ \mathbf{x} &\mapsto \sigma(\mathbf{x}) = \mathbf{H}^t \mathbf{x}\end{aligned}$$

Le *code linéaire* associé est le sous-espace vectoriel de \mathbb{F}_2^n

$$\{\mathbf{x} \mid \sigma(\mathbf{x}) = \mathbf{0}\}$$

On dit que \mathbf{H} est une matrice de contrôle du code C .
Sa dimension est

$$\dim(C) = k = n - r.$$

Décodage par syndrome

Mot de code émis : \mathbf{c}

Vecteur reçu : \mathbf{x}

$$\mathbf{x} = \mathbf{c} + \mathbf{e}$$

où \mathbf{e} vecteur (erreur) de poids t (t symboles erronés).

$$\sigma(\mathbf{x}) = \sigma(\mathbf{c}) + \sigma(\mathbf{e}) = \sigma(\mathbf{e}) = \mathbf{s}.$$

Si \mathbf{s} est l'**unique** syndrome d'un vecteur de poids $\leq t$, alors le décodeur qui cherche le vecteur de syndrome \mathbf{s} de poids minimum trouve le vecteur erreur \mathbf{e} .

Donc : Si l'application

$$\begin{aligned} B_t(0) &\rightarrow \{0, 1\}^r \\ \mathbf{x} &\mapsto \sigma(\mathbf{x}) \end{aligned}$$

est **injective**, alors le code peut corriger une configuration arbitraire de t erreurs.

Théorème de Shannon pour les codes linéaires

On choisit \mathbf{H} , la matrice $r \times n$ définissant le code C *au hasard* (uniformément), avec $r = n(h(p) + \varepsilon)$.

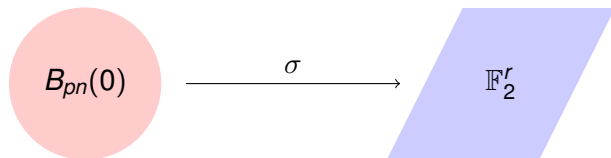
$$R = \frac{k}{n} = 1 - h(p) - \varepsilon.$$

Fonction syndrome :

$$\begin{aligned} B_{pn}(0) &\rightarrow \{0, 1\}^r \\ \mathbf{e} &\mapsto \mathbf{H}^t \mathbf{e} \\ |B_{pn}(0)| &\approx \binom{n}{pn} \approx 2^{nh(p)} \\ 2^r &= 2^{n(h(p)+\varepsilon)} \end{aligned}$$

et la fonction σ sur $B_{pn}(0)$ est **presque** injective.

Le théorème de Shannon et la fonction syndrome



Vecteur reçu \mathbf{x} de syndrome \mathbf{s} .

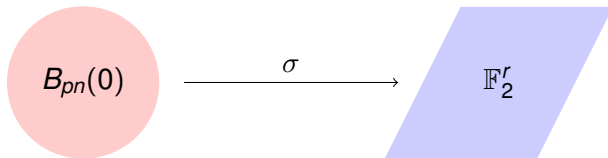
Le décodeur qui choisit $f(\mathbf{x}) =$ le vecteur de syndrome \mathbf{s} de plus petit poids trouve **presque** toujours le bon vecteur erreur \mathbf{e} .

Se débarrasser du « presque » ?

Le vecteur erreur \mathbf{e} doit avoir un poids $t \leq d_{\min}/2$,

$$\mathbf{c} \xrightarrow{\leq t} \mathbf{x} \xrightarrow{\leq t} \mathbf{c}'$$

$$\sigma(\mathbf{x}) = \sigma(\mathbf{x} - \mathbf{c}) = \sigma(\mathbf{x} - \mathbf{c}').$$



Si $|B_{pn}(0)| \geq 2^{\varepsilon n} 2^r$, alors presque sûrement il y a un élément de syndrome nul dans la boule, i.e. un mot de code.

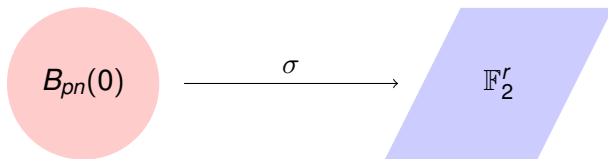
d_{\min} typique d'un code aléatoire : $\frac{k}{n} \geq 1 - h\left(\frac{d_{\min}}{n}\right)$.

Se débarrasser du « presque » ?

Le vecteur erreur \mathbf{e} doit avoir un poids $t \leq d_{\min}/2$,

$$\mathbf{c} \xrightarrow{\leq t} \mathbf{x} \xrightarrow{\leq t} \mathbf{c}'$$

$$\sigma(\mathbf{x}) = \sigma(\mathbf{x} - \mathbf{c}) = \sigma(\mathbf{x} - \mathbf{c}').$$



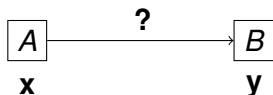
Si $|B_{pn}(0)| \geq 2^{\varepsilon n} 2^r$, alors presque sûrement il y a un élément de syndrome nul dans la boule, i.e. un mot de code.

d_{\min} typique d'un code aléatoire : $\frac{k}{n} \geq 1 - h\left(\frac{d_{\min}}{n}\right)$.

Nombre d'erreurs corrigibles divisé par 2 !

Autres utilisations de la fonction syndrome

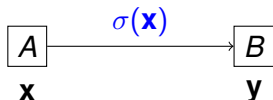
A dispose d'un vecteur \mathbf{x} et B de \mathbf{y} avec $d(\mathbf{x}, \mathbf{y}) \leq t$.



Comment A et B *réconcilient* leur n -uples en transmettant le moins de bits possibles ?

Autres utilisations de la fonction syndrome

A dispose d'un vecteur \mathbf{x} et B de \mathbf{y} avec $d(\mathbf{x}, \mathbf{y}) \leq t$.



Comment A et B *réconcilient* leur n -uples en transmettant le moins de bits possibles ?

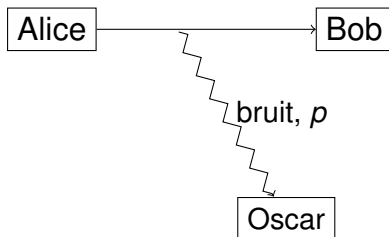
Solution : A transmet à B le syndrome $\sigma(\mathbf{x})$ associé à un code t -correcteur.

B calcule

$$\sigma(\mathbf{x}) + \sigma(\mathbf{y}) = \sigma(\mathbf{x} + \mathbf{y}) = \sigma(\mathbf{e})$$

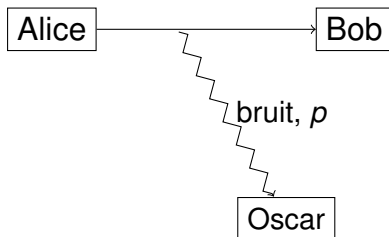
où $\text{poids}(\mathbf{e}) \leq t$.

Canal à jarretière (wiretap channel)



Comment Alice peut-elle communiquer un secret à Bob sans concéder le moindre bit d'information à Oscar ? Quelle est la taille maximale du secret si Alice transmet n symboles ?

Canal à jarretière (wiretap channel)

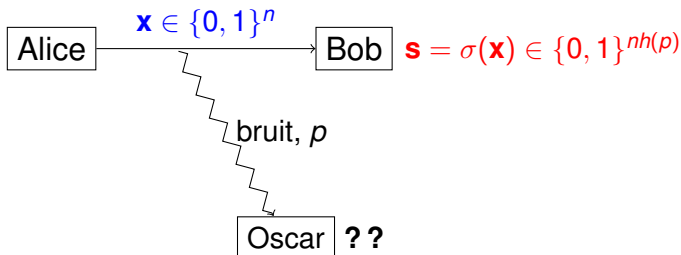


Comment Alice peut-elle communiquer un secret à Bob sans concéder le moindre bit d'information à Oscar ? Quelle est la taille maximale du secret si Alice transmet n symboles ?

Indice : si Alice transmet \mathbf{x} constitué de n symboles, Oscar obtient au mieux $n(1 - h(p))$ bits d'information **sur \mathbf{x}** .

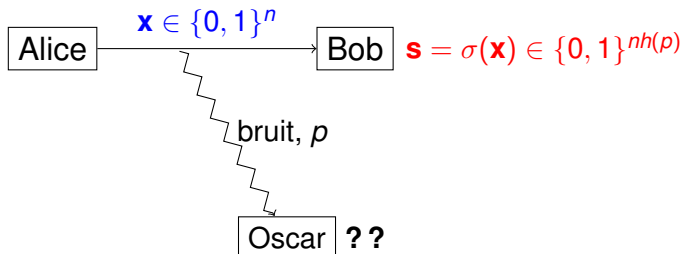
On peut donc espérer que transmettre \mathbf{x} permette de transmettre un secret de $nh(p)$ bits.

Canal à jarretière, transmission du secret



Solution : Alice et Bob se mettent d'accord sur un code linéaire aléatoire de dimension $k = n(1 - h(p))$. Pour transmettre un secret $\mathbf{s} \in \{0, 1\}^r$, $r = n - k = nh(p)$, Alice choisit un vecteur \mathbf{x} aléatoire parmi tous ceux de syndrome \mathbf{s} .

Canal à jarretière, transmission du secret

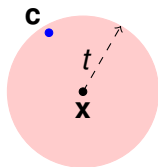


Solution : Alice et Bob se mettent d'accord sur un code linéaire aléatoire de dimension $k = n(1 - h(p))$. Pour transmettre un secret $\mathbf{s} \in \{0, 1\}^r$, $r = n - k = nh(p)$, Alice choisit un vecteur \mathbf{x} aléatoire parmi tous ceux de syndrome \mathbf{s} .

- Oscar reçoit $\mathbf{y} = \mathbf{x} + \mathbf{e}$. On a $\sigma(\mathbf{e})$ de loi presque uniforme
- $H(\mathbf{s} | \mathbf{y}) = H(\mathbf{s} | \mathbf{s} + \sigma(\mathbf{e}))$.

Oscar : presque zéro bit d'information sur \mathbf{s} .

Problème algorithmique du décodage



Comment trouver, à partir du vecteur reçu \mathbf{x} , le mot code \mathbf{c} le plus proche de \mathbf{x} ?

Est-ce que cela aide de savoir que $d(\mathbf{c}, \mathbf{x}) \leq t$?

Du binaire au q -aire

Transformer un train de symboles binaires

0010110110100011 ...

Du binaire au q -aire

Transformer un train de symboles binaires

0010110110100011 ...

en suite d'octets

00101101	10100011	...
----------	----------	-----

que l'on traite comme des éléments de \mathbb{F}_{256} . Code sur \mathbb{F}_q .

Erreur sur un bit \rightarrow erreur sur un octet.

Codes de Reed-Solomon

Encodage : on fixe $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}_q$, distincts.

$$\begin{aligned} \mathbb{F}_q^k &\rightarrow \mathbb{F}_q^n \\ (a_0, a_1, \dots, a_{k-1}) &\mapsto (f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)) \end{aligned}$$

où $f(X) = a_0 + a_1X + \dots + a_{k-1}X^{k-1}$.

Le code de *Reed Solomon* associé est :

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

Codes de Reed-Solomon

Encodage : on fixe $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}_q$, distincts.

$$\begin{aligned} \mathbb{F}_q^k &\rightarrow \mathbb{F}_q^n \\ (a_0, a_1, \dots, a_{k-1}) &\mapsto (f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)) \end{aligned}$$

où $f(X) = a_0 + a_1X + \dots + a_{k-1}X^{k-1}$.

Le code *de Reed Solomon* associé est :

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

$$\dim C = k$$

et $\mathbf{c}_f - \mathbf{c}_g$ a au plus $k - 1$ zéros dans \mathbb{F}_q , donc

$$d_{\min} = n - k + 1.$$

Distance minimale optimale.

Correction d'erreurs, approche classique

vecteur émis : (c_1, \dots, c_n) , $c_i = f(\alpha_i)$.

vecteur reçu : (r_1, \dots, r_n) .

t symboles erronés, $t = \#\{i, r_i \neq c_i\}$.

polynôme localisateur d'erreur :

$$E(X) = \prod_{i, r_i \neq c_i} (X - \alpha_i).$$

Équation clé :

$$\forall i, \quad f(\alpha_i)E(\alpha_i) = r_i E(\alpha_i).$$

Si $Q(X) = f(X)E(X)$ on constate : si $t < (n - k)/2$,

- 1 $\deg E \leq t$, $\deg Q < k + t$, $E(X)$ est unitaire,
- 2 $\forall i, \quad Q(\alpha_i) = r_i E(\alpha_i)$.

Correction d'erreurs, approche classique

- 1 $\deg E \leq t$, $\deg Q < k + t$, $E(X)$ est unitaire,
- 2 $\forall i, Q(\alpha_i) = r_i E(\alpha_i)$.

Si un *autre* couple solution $(Q_1(X), E_1(X))$, on a :

$$\forall i, Q(\alpha_i)E_1(\alpha_i)r_i = r_i E(\alpha_i)Q_1(\alpha_i)$$

donc :

$$\forall i, Q(\alpha_i)E_1(\alpha_i) = E(\alpha_i)Q_1(\alpha_i)$$

Mais (1) et $t < (n - k)/2$ implique $\deg(QE_1 - EQ_1) < n$, donc $QE_1 - EQ_1 = 0$ et

$$\frac{Q_1(X)}{E_1(X)} = \frac{Q(X)}{E(X)} = f(X).$$

Correction d'erreurs, approche classique

- 1 $\deg E \leq t$, $\deg Q < k + t$, $E(X)$ est unitaire,
- 2 $\forall i, Q(\alpha_i) = r_i E(\alpha_i)$.

Si un *autre* couple solution $(Q_1(X), E_1(X))$, on a :

$$\forall i, Q(\alpha_i)E_1(\alpha_i)r_i = r_i E(\alpha_i)Q_1(\alpha_i)$$

donc :

$$\forall i, Q(\alpha_i)E_1(\alpha_i) = E(\alpha_i)Q_1(\alpha_i)$$

Mais (1) et $t < (n - k)/2$ implique $\deg(QE_1 - EQ_1) < n$, donc $QE_1 - EQ_1 = 0$ et

$$\frac{Q_1(X)}{E_1(X)} = \frac{Q(X)}{E(X)} = f(X).$$

Les coefficients de (2) vérifient un système d'équations linéaires !

Ubiquité des codes de Reed-Solomon

- Disques compacts (CD)
- DVD
- enregistrements magnétiques (disques durs)
- téléphonie sans fil 3G (IS-2000)
- réseaux de fibre optique (ITU-T G.795)
- ADSL (ITU-T G. 992.1)
- systèmes sans fil large bande (MAN, IEEE 802.6)
- stations Intelsat (IESS-308)
- diffusion numérique par satellite (ETS 300-421S, ETS 300-429)
- exploration spatiale (sondes NASA)
- ...

Codes de Reed-Solomon et cryptographie

Procédé de partage de secret. Soit $s \in \mathbb{F}_q$ un secret à *partager* entre n utilisateurs.

On choisit aléatoirement, uniformément un polynôme

$$f(X) = s + a_1X + a_2X^2 + \dots + a_{k-1}X^{k-1}$$

et on distribue $f(\alpha_1), \dots, f(\alpha_n)$. L'utilisateur i a $(\alpha_i, f(\alpha_i))$.

- Si k utilisateurs quelconques se réunissent, ils déterminent s (interpolation).
- Si $k - 1$ utilisateurs quelconques se réunissent, ils ont 0 bit d'information sur s .

Les limites du décodage traditionnel

On cherche à obtenir $f(X)$ comme solution *unique* d'un système algébrique, par exemple :

$$f(X) = \frac{Q(X)}{E(X)}.$$

Impossible dans ces conditions de corriger plus de $(n - k)/2$ ($d_{\min}/2$) erreurs.

Pourtant, en principe, le mot de code le plus proche est presque toujours le bon jusqu'à un nombre d'erreurs proche de

$$n - k.$$

Comment faire ?

L'idée de Sudan (1997)

Chercher la *liste* des mots de code à distance $\leq t$ d'un quelconque vecteur.

Problème simplifié : soit un vecteur (r_1, \dots, r_n) obtenu en «mélangeant» deux mots $(f(\alpha_1) \dots f(\alpha_n))$ et $(g(\alpha_1) \dots g(\alpha_n))$,

$$r_i = f(\alpha_i) \quad \text{ou} \quad r_i = g(\alpha_i).$$

Comment retrouver f , ou g à partir de (r_i) ?

L'idée de Sudan (1997)

Chercher la *liste* des mots de code à distance $\leq t$ d'un quelconque vecteur.

Problème simplifié : soit un vecteur (r_1, \dots, r_n) obtenu en «mélangeant» deux mots $(f(\alpha_1) \dots f(\alpha_n))$ et $(g(\alpha_1) \dots g(\alpha_n))$,

$$r_i = f(\alpha_i) \quad \text{ou} \quad r_i = g(\alpha_i).$$

Comment retrouver f , ou g à partir de (r_i) ?

Remarque clé : soit

$$Q(X, Y) = (Y - f(X))(Y - g(X)).$$

Si on peut trouver $Q(X, Y)$, il suffit de le factoriser. On a :

$$\forall i \quad Q(\alpha_i, r_i) = 0.$$

$$Q(X, Y) = Y^2 - S(X)Y + P(X).$$

$\deg S(X) < k$, $\deg P(X) < 2k - 1$. Les coefficients de $Q(X, Y)$ se trouvent en résolvant un **système linéaire**.

L'algorithme de Sudan

Et si la solution de

$$\forall i \quad Q(\alpha_i, r_i) = 0.$$

n'est pas la bonne $((Y - f(X))(Y - g(X)))$?

On a l'argument : si $f(\alpha_j) = r_j$ pour au moins $2k - 1$ valeurs, alors $Q(X, f(X)) = 0$ (s'annule $2k - 1$ fois, degré $< 2k - 1$).

Donc

$$(Y - f(X)) \mid Q(X, Y).$$

Marche jusqu'à $k = n/4$. On peut retrouver plus d'une solution (f ou g).

L'algorithme de Sudan

Et si la solution de

$$\forall i \quad Q(\alpha_i, r_i) = 0.$$

n'est pas la bonne $((Y - f(X))(Y - g(X)))$?

On a l'argument : si $f(\alpha_j) = r_j$ pour au moins $2k - 1$ valeurs, alors $Q(X, f(X)) = 0$ (s'annule $2k - 1$ fois, degré $< 2k - 1$).

Donc

$$(Y - f(X)) \mid Q(X, Y).$$

Marche jusqu'à $k = n/4$. On peut retrouver plus d'une solution (f ou g).

On peut complètement oublier le but initial (trouver $(Y - f(X))(Y - g(X))$). N'importe quel $Q(X, Y)$ convient.

Algorithme de Sudan, cas général

Vecteur émis : $(f(\alpha_1), \dots, f(\alpha_n))$, $\deg f < k$.

Vecteur reçu : $(r_1 \dots r_n)$, $r_i = f(\alpha_i)$, sauf pour au plus t valeurs de i . #erreurs $\leq t$.

Trouver $Q(X, Y) = P_m(X)Y^m + P_{m-1}(X)Y^{m-1} + \dots + P_0(X)$ tel que

$$\forall i \quad Q(\alpha_i, r_i) = 0$$

en résolvant le système linéaire satisfait par les coefficients de (P_i) . Solution non-triviale garantie si #coefficients $> n$.

S'assurer que si $\deg f < k$ alors $\deg Q(X, f(X)) < n - t$, donc $Q(X, f(X)) = 0$.

Factoriser $Q(X, Y)$. Mieux, trouver les facteurs de $Q(X, Y)$ de la forme $Y - f(X)$. Liste des mots à distance $\leq t$ de $(r_1 \dots r_n)$.

Algorithme de Sudan et améliorations

Correction de t erreurs :

- $\frac{t}{n} = (1 - R)/2$ (Reed Solomon, 1960, Berlekamp-Massey)
- $\frac{t}{n} = 1 - \sqrt{2R}$ (Sudan, 1997)
- $\frac{t}{n} \rightarrow 1 - \sqrt{R}$ (Guruswami et Sudan, 1999)
- ...
- $\frac{t}{n} \rightarrow 1 - R$ (Guruswami et Rudra, 2007).

Références

- G. Battail, *Théorie de l'Information*, Masson, 1997.
- T. M. Cover et J. A. Thomas, *Elements of Information Theory*, Wiley, 2005.
- J. H. van Lint, *An Introduction to coding theory*, Springer, 1998.
- F. J. MacWilliams et N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, 2003.
- R. M. Roth, *Introduction to coding theory*, Cambridge University Press, 2006.
- La page de M. Sudan :
<http://people.csail.mit.edu/madhu/>
- G. Zémor, *Cours de cryptographie*, Cassini, 2000.