Les codes algébriques principaux et leur décodage

Daniel Augot

INRIA Saclay-Île de France et École polytechnique

Journées nationales du calcul formel
Luminy, mai 2010

Plan

Codage, Shannon et Hamming

Codes de Reed-Solomon, de Reed et Muller, de Goppa géométriques

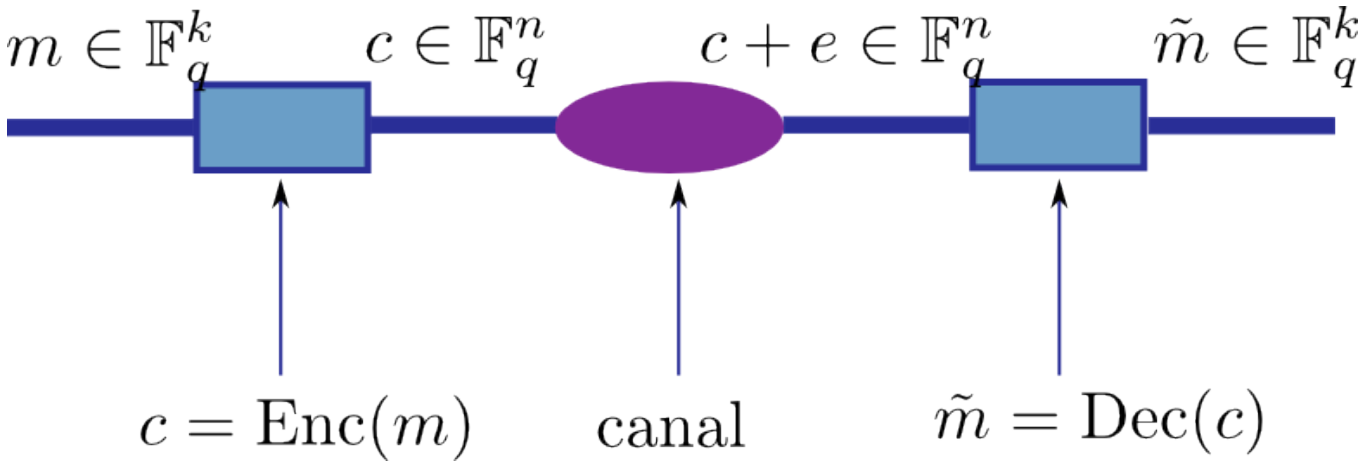
Décodage par syndrome des codes de Reed-Solomon, des codes de Goppa

Décodage par interpolation des codes de Reed-Solomon et des codes géométriques : Sudan

Plan

Codes correcteurs

Canal de transmission (cas particulier). Schéma de codage, $k < n$.



Point de vue de Shannon : on veut que la quantité suivante soit petite

$$\min_{\mathrm{Enc},\mathrm{Dec}} \left\{ \max_{m \in \mathbb{F}_q^k} \left(\Pr_e[\mathrm{Dec}(\mathrm{Enc}(m) + e) \neq m] \right) \right\},$$

et que $R = \frac{k}{n} \in [0, 1]$ soit grand. R est le taux de transmission.

Début de la théorie de l'information

Théorème (Shannon 1948)

Pour tout canal discret sans mémoire, il existe une quantité $C \in [0,1]$, la capacité du canal, telle qu'on puisse communiquer de manière fiable avec un taux $R < C$.
Plus spécifiquement, il existe une séquence de codes tels que

1. *le taux de transmission est $R < C$;*
2. *leur longueur tend vers l'infini ;*
3. *la probabilité d'erreur de décodage tend vers 0.*

Complétement non constructif.

Réciproque

Pour toute famille de codes telle que la probabilité maximale de décodage tend vers 0, on a : $\limsup k/n < C$.

Codes correcteurs

Point de vue de Hamming (1950) : la distance de Hamming entre deux mots est

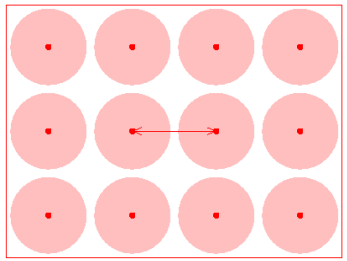
$$d(x, y) = |\{i;\ x_i \neq y_i\}|\,.$$

Un code C est l'image de sa fonction d'encodage :

$$\left\{\mathrm{Enc}(m), \quad m \in \mathbb{F}_q^k\right\}.$$

La distance minimale est

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



Il y a au plus un mot de code à distance $t = \lfloor \frac{d-1}{2} \rfloor$ du mot reçu.
On parle de code t-correcteur. Notation : code $[n, k, d]_q$.

Shannon versus Hamming

Canal q-aire symétrique : $\Pr(\hat{y}_i \neq y_i) = p$.

- Shannon. Approche probabiliste (On parle du *canal aléatoire*) :

$$\left\{ \max_{m \in \mathbb{F}_q^k} \left(\Pr_e[\mathrm{Dec}(\mathrm{Enc}(m) + e) \neq m] \right) \right\} \text{ doit être petit.}$$

  $\Longrightarrow$ décodage itératif : turbo-codes, codes LDPC, etc.

- Hamming. Approche déterministe (On parle du *canal adversaire*) :

  Tout motif de poids inférieur à $t = \lfloor \frac{d-1}{2} \rfloor$ *est corrigé.*

  Nombre d'erreurs limités, mais pire cas possible.
  $\Longrightarrow$ codes algébriques, décodage déterministe.

À quoi servent principalement les codes correcteurs d'erreur

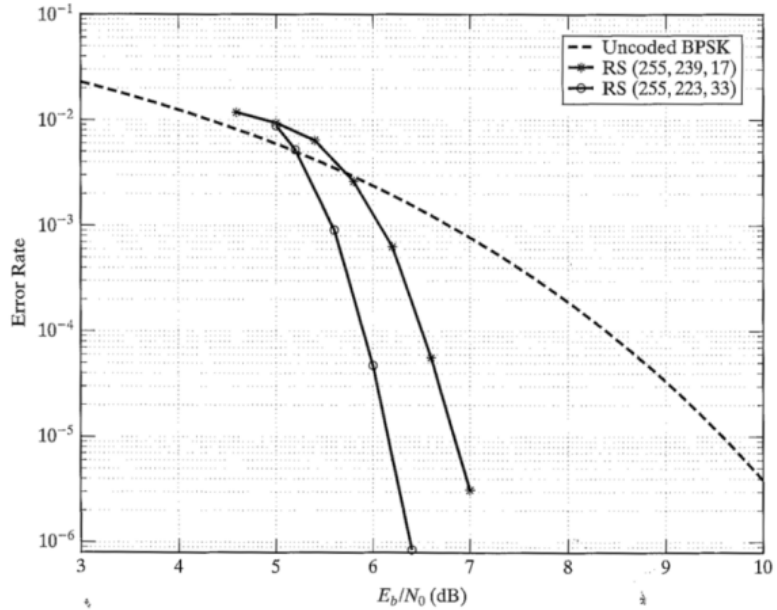


FIG. 1: Taux d'erreur par bit après décodage en fonction du rapport signal à bruit (échelles logarithmiques)

- À améliorer la qualité d'une communication donnée ;
- à diminuer le rapport signal à bruit, pour un taux d'erreur cible : « gain de codage ».

À quoi servent principalement les codes correcteurs d'erreur

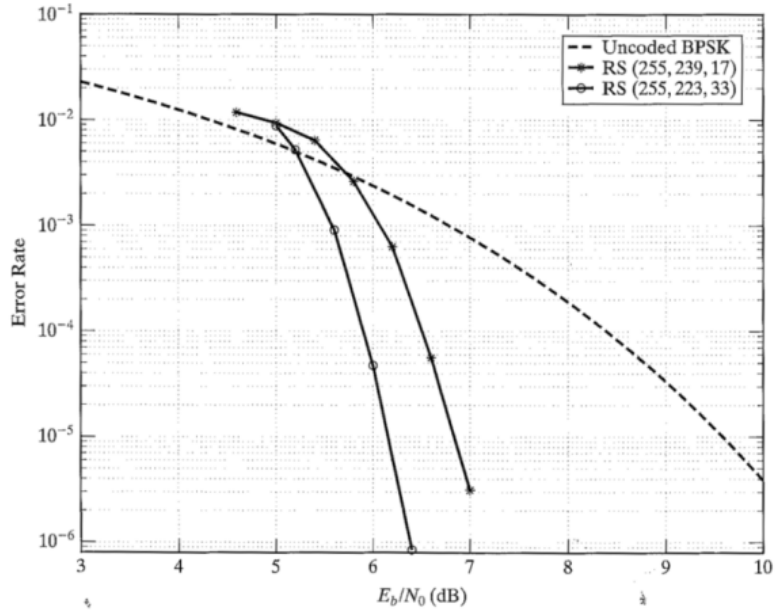


FIG. 1: Taux d'erreur par bit après décodage en fonction du rapport signal à bruit (échelles logarithmiques)

- À améliorer la qualité d'une communication donnée ;
- à diminuer le rapport signal à bruit, pour un taux d'erreur cible : « gain de codage ».
  $\Longrightarrow$ diminution de la puissance, miniaturisation, autonomie.

Plan

Codes de Reed-Solomon

On considère $\mathbb{F}_q$, par exemple $\mathbb{F}_{256}$.
Soient $x_1, \ldots, x_n \in \mathbb{F}_q$, deux à deux distincts, avec $n \leqslant q$,

Définition

La fonction d'évaluation associée à $x_1, \ldots, x_n$ est

$$\begin{array}{rccc} \mathrm{ev}: & \mathbb{F}_q[x] & \to & \mathbb{F}_q^n \\ & f & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

Définition

Soit $L_k = \{f \in \mathbb{F}_q[x];\ \deg f < k\}$. Le code de Reed-Solomon de support $x_1, \ldots, x_n$, de dimension k est

$$C = \mathrm{ev}(L_k).$$

On a $d = n - k + 1$. On parle de code $[n, k, d]_q$.

Dualité

- Sur $\mathbb{F}_q^n$, on définit le « produit scalaire standard » :

$$x \cdot y = \sum x_i y_i.$$

- Pour un code C, on a donc le code dual

$$C^\perp = \{y;\ y \cdot x = 0;\ \forall x \in C\}.$$

- Soit $b_1, \ldots, b_k$ une base d'un code C.
  La matrice formée des lignes des coefficients des b_i est une matrice génératrice de C.
- Soit $b_1^\perp, \ldots, b_{n-k}^\perp$ une base du dual d'un code C.
  La matrice formée des lignes des coefficients des $b_i^\perp$ est une matrice de parité de C.

Cas des codes de Reed-Solomon

Proposition

Soit α une racine n-ième de l'unité.
Soit $\mathrm{RS}(\alpha, k)$ *le code de Reed-Solomon de support* $\alpha^0, \cdots, \alpha^{n-1}$, *de dimension* k. *Alors*

$$\mathrm{RS}(\alpha, k)^{\perp} = \mathrm{RS}(\alpha, n-k).$$

Cette dualité permettra de construire deux familles d'algorithmes de décodage :

1. « décodage par syndrome », pour un code dual de code d'évaluation ;
2. « décodage par interpolation », pour un code d'évalution.

Cas des codes de Reed-Solomon

Proposition

Soit α une racine n-ième de l'unité.
Soit $\mathrm{RS}(\alpha, k)$ *le code de Reed-Solomon de support* $\alpha^0, \cdots, \alpha^{n-1}$, *de dimension* k. *Alors*

$$\mathrm{RS}(\alpha, k)^{\perp} = \mathrm{RS}(\alpha, n-k).$$

Cette dualité permettra de construire deux familles d'algorithmes de décodage :

1. « décodage par syndrome », pour un code dual de code d'évaluation ; Première partie de l'exposé.
2. « décodage par interpolation », pour un code d'évalution.

Cas des codes de Reed-Solomon

Proposition

Soit α une racine n-ième de l'unité.
Soit $\mathrm{RS}(\alpha, k)$ *le code de Reed-Solomon de support* $\alpha^0, \cdots, \alpha^{n-1}$, *de dimension* k. *Alors*

$$\mathrm{RS}(\alpha, k)^{\perp} = \mathrm{RS}(\alpha, n-k).$$

Cette dualité permettra de construire deux familles d'algorithmes de décodage :

1. « décodage par syndrome », pour un code dual de code d'évaluation ; Première partie de l'exposé.
2. « décodage par interpolation », pour un code d'évalution. Deuxième partie de l'exposé.

Borne de Singleton

Théorème

Tout code C de paramètres $[n, k, d]$ vérifie $d \leqslant n - k + 1$.

Définition

Un code $[n, k, d]$ vérifiant $d = n - k + 1$ est dit MDS *(Maximum Distance Separable).*

Les codes de Reed-Solomon sont donc MDS, et on verra qu'on peut décoder efficacement jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.

Borne de Singleton

Théorème

Tout code C de paramètres $[n, k, d]$ vérifie $d \leqslant n - k + 1$.

Définition

Un code $[n, k, d]$ vérifiant $d = n - k + 1$ est dit MDS *(Maximum Distance Separable).*

Les codes de Reed-Solomon sont donc MDS, et on verra qu'on peut décoder efficacement jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.

- Fin de la théorie des codes ?

Borne de Singleton

Théorème

Tout code C de paramètres $[n, k, d]$ vérifie $d \leqslant n - k + 1$.

Définition

Un code $[n, k, d]$ vérifiant $d = n - k + 1$ est dit MDS *(Maximum Distance Separable).*

Les codes de Reed-Solomon sont donc MDS, et on verra qu'on peut décoder efficacement jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.

- Fin de la théorie des codes ?
- Pour définir un code de Reed-Solomon, il faut $x_1, \ldots, x_n \in \mathbb{F}_q$ distincts.

Borne de Singleton

Théorème

Tout code C de paramètres $[n, k, d]$ vérifie $d \leqslant n - k + 1$.

Définition

Un code $[n, k, d]$ vérifiant $d = n - k + 1$ est dit MDS *(Maximum Distance Separable).*

Les codes de Reed-Solomon sont donc MDS, et on verra qu'on peut décoder efficacement jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.

- Fin de la théorie des codes ?
- Pour définir un code de Reed-Solomon, il faut $x_1, \ldots, x_n \in \mathbb{F}_q$ distincts.
- Donc $n \leqslant q$!

Borne de Singleton

Théorème

Tout code C de paramètres $[n, k, d]$ vérifie $d \leqslant n - k + 1$.

Définition

Un code $[n, k, d]$ vérifiant $d = n - k + 1$ est dit MDS *(Maximum Distance Separable).*

Les codes de Reed-Solomon sont donc MDS, et on verra qu'on peut décoder efficacement jusqu'à $t = \lfloor \frac{d-1}{2} \rfloor$ erreurs.

- Fin de la théorie des codes ?
- Pour définir un code de Reed-Solomon, il faut $x_1, \ldots, x_n \in \mathbb{F}_q$ distincts.
- Donc $n \leqslant q$!
- Or le théorème de Shannon dit qu'il faut des codes longs.

Codes MDS

- Est-il possible d'avoir des codes MDS avec $n > q$?
- En « étendant » les codes de Reed-Solomon, on arrive à $n = q + 1$.

Conjecture (Conjecture MDS)

Tous les codes MDS vérifient $n \leqslant q + 1$, *sauf si* $k = 3$ *ou* $k = q - 1$, *avec* q *pair, auquel cas on a* $n \leqslant q + 2$.

Or le cas le plus intéressant est q petit, voire $q = 2$. Ou aussi $q = 256$.

Codes MDS pour « le canal à effacement »

Une code MDS est tel que toute sous-matrice de taille k d'une matrice génératrice $k \times n$ est de rang k. « Matrice MDS ».
$\Longrightarrow$ recherche de codes MDS « à bas coût », near MDS sur des alphabets plus petits, etc.

Codes MDS

- Est-il possible d'avoir des codes MDS avec $n > q$?
- En « étendant » les codes de Reed-Solomon, on arrive à $n = q + 1$.

Conjecture (Conjecture MDS)

Tous les codes MDS vérifient $n \leqslant q + 1$, *sauf si* $k = 3$ *ou* $k = q - 1$, *avec* q *pair, auquel cas on a* $n \leqslant q + 2$.

Or le cas le plus intéressant est q petit, voire $q = 2$. Ou aussi $q = 256$.

Codes MDS pour la perte de paquets

Une code MDS est tel que toute sous-matrice de taille k d'une matrice génératrice $k \times n$ est de rang k. « Matrice MDS ».
$\Longrightarrow$ recherche de codes MDS « à bas coût », near MDS sur des alphabets plus petits, etc.

Bornes élémentaires pour des alphabets « petits »

Soit $B_q(n,t)$ la boule de rayon de t centrée en 0. Son volume est

$$V_q(n,t) = \sum_{i=0}^{t} (q-1)^i \binom{n}{i}.$$

Proposition (Borne de Hamming)

Tout code $[n,k,d]_q$ *vérifie :*

$$q^k V_q(n,t) \leqslant q^n, \quad \text{où } t = \lfloor \frac{d-1}{2} \rfloor.$$

Proposition (Borne de Gilbert-Varshamov)

Si

$$q^k V_q(n,d-1) < q^n,$$

alors il existe un code $[n,k,d]_q$.

Bornes élémentaires pour des alphabets « petits »

Soit $B_q(n,t)$ la boule de rayon de t centrée en 0. Son volume est

$$V_q(n,t) = \sum_{i=0}^{t} (q-1)^i \binom{n}{i}.$$

Proposition (Borne de Hamming)

Tout code $[n,k,d]_q$ *vérifie :*

$$q^k V_q(n,t) \leqslant q^n, \quad \text{où } t = \lfloor \frac{d-1}{2} \rfloor.$$

Partitionnement de l'espace.

Proposition (Borne de Gilbert-Varshamov)

Si

$$q^k V_q(n,d-1) < q^n,$$

alors il existe un code $[n,k,d]_q$.

Bornes élémentaires pour des alphabets « petits »

Soit $B_q(n,t)$ la boule de rayon de t centrée en 0. Son volume est

$$V_q(n,t) = \sum_{i=0}^{t} (q-1)^i \binom{n}{i}.$$

Proposition (Borne de Hamming)

Tout code $[n,k,d]_q$ *vérifie :*

$$q^k V_q(n,t) \leqslant q^n, \quad \text{où } t = \lfloor \frac{d-1}{2} \rfloor.$$

Partitionnement de l'espace.

Proposition (Borne de Gilbert-Varshamov)

Si

$$q^k V_q(n,d-1) < q^n,$$

alors il existe un code $[n,k,d]_q$*. Algorithme glouton.*

Formes asymptotiques

Lemme

Lorsque le rapport t/n est constant, on a

$$\frac{\log_q V_q(n,t)}{n} \sim H_q(t/n),$$

où $H_q(x)$ est la fonction d'entropie q-aire :

$$H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x).$$

Proposition (Bornes de Hamming et de Gilbert-Varshamov asymptotiques)

On a donc, pour $R = \limsup \frac{k}{n}$, et $\delta = \frac{d}{n}$ fixé :

$$1 - H_q(\delta) \leqslant R \leqslant 1 - H_q(\delta/2).$$

Courbes

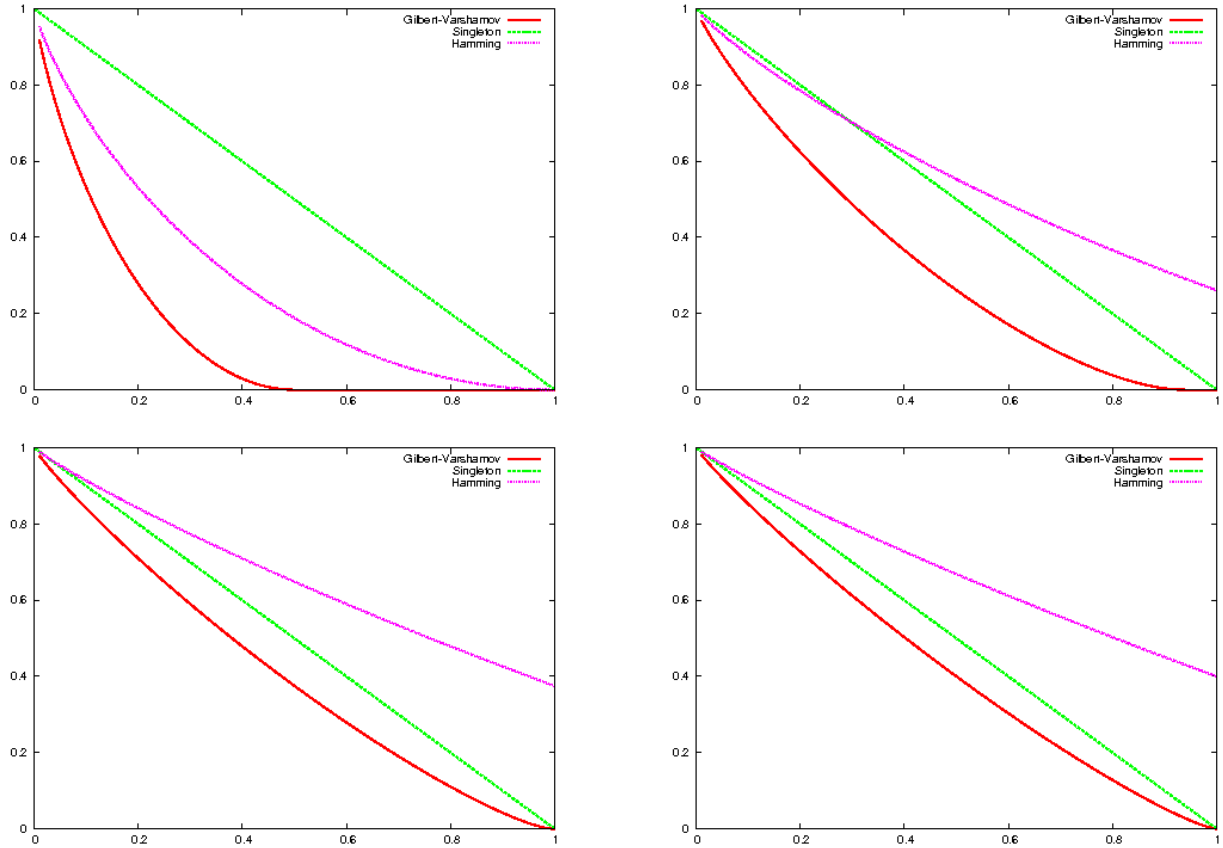


FIG. 2: Taux R en fonction de $\delta = d/n$ pour $q = 2, 16, 256, 1024$.

Les impossibilités

1. Un code aléatoire assez long sera sur la borne de Varshamov-Gilbert, avec probabilité tendant exponentiellement vers 1 !
2. Mais on ne saura pas déterminer sa distance minimale !

   A. Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In STOC '97.

3. On ne saura pas le décoder !

   E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. IEEE Trans. on Inf. Theory, *1978.*

Les impossibilités

1. Un code aléatoire assez long sera sur la borne de Varshamov-Gilbert, avec probabilité tendant exponentiellement vers 1 !
2. Mais on ne saura pas déterminer sa distance minimale !

   A. Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In STOC '97.

3. On ne saura pas le décoder !

   E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. IEEE Trans. on Inf. Theory, *1978.*

4. Même avec précalcul !

   J. Bruck and M. Naor. The hardness of decoding linear codes with preprocessing. IEEE Trans. on Inf. Theory, *1990*

Tentatives

Définition

Une famille de codes $[n_i, k_i, d_i]_q$, *avec* $n_i \to \infty$ *est* *asymptotiquement bonne*, *si*

$$\left(\limsup \frac{k_i}{n_i}\right) > 0 \text{ et } \left(\limsup \frac{d_i}{n_i}\right) > 0.$$

Essais

À partir d'un code de Reed-Solomon sur $\mathbb{F}_{2^m}$.

1. On considère $\mathrm{RS}(\alpha, k) \cap \mathbb{F}_2^n$: c'est le code « BCH » de distance construite $d = n - k + 1$.

Théorème : Les codes BCH binaires sont asymptotiquement mauvais.

Tentatives

Définition

Une famille de codes $[n_i, k_i, d_i]_q$*, avec* $n_i \to \infty$ *est asymptotiquement bonne, si*

$$\left(\limsup \frac{k_i}{n_i}\right) > 0 \text{ et } \left(\limsup \frac{d_i}{n_i}\right) > 0.$$

Essais

À partir d'un code de Reed-Solomon sur $\mathbb{F}_{2^m}$.

1. On considère $\mathrm{RS}(\alpha, k) \cap \mathbb{F}_2^n$: c'est le code « BCH » de distance construite $d = n - k + 1$.

Théorème : Les codes BCH binaires sont asymptotiquement mauvais.

2. « Déploiement » d'un code de Reed-Solomon $[N, K, D]_{2^m}$ en un code binaire $[mN, mK, d \geqslant D]_2$

Théorème : Ces codes contiennent les BCH. $\Longrightarrow$ Ils sont asymptotiquement mauvais.

Codes de Reed-Muller

Une tentative naturelle est de considérer plusieurs variables.

- On note $n = q^m$, et on énumère $\mathbb{F}_q^m = \{P_1, \ldots, P_n\}$;
- la fonction d'évaluation associée aux points $\{P_1, \ldots, P_n\}$ est

$$\begin{array}{rccc} \mathrm{ev} : & \mathbb{F}_q[x_1, \ldots, x_m] & \longrightarrow & \mathbb{F}_q^n \\ & f & \longmapsto & (f(P_1), \ldots, f(P_n)) \end{array}$$

Définition (Codes de Reed-Muller généralisés)

On considère l'espace, pour $0 \leqslant r \leqslant m(q-1)$,

$$L_r = \{f \in \mathbb{F}_q[X_1, \ldots, X_m];\ \deg f \leqslant r\}.$$

Le code de Reed-Muller q-aire d'ordre r en m variables est $\mathrm{RM}_q(r, m) = \mathrm{ev}(L_r)$.

Code de longueur $n = q^m$, de longueur exponentielle en m, à q fixé.

Paramètres des codes de Reed-Muller

Proposition (Lemme de Schwartz-Zippel)

Soit $f \in \mathbb{F}_q[X_1, \ldots, X_m]$ de degré total au plus égal à r. Alors le nombre de zéros de f sur $\mathbb{F}_q^m$ est borné par rq^{m-1}.

Corollaire

Pour $r \leqslant q-1$, le code de Reed-Muller d'ordre r a pour dimension $\binom{r+m}{m}$ et $(q-r)q^{m-1}$ comme distance minimale.

Il a pour taux de transmission

$$R = \frac{k}{n} \leqslant \left(\frac{r+1}{q}\right)^m,$$

et, pour distance minimale relative :
$\delta = \frac{d}{n} = (q-r)q^{m-1}/q^m = 1 - \frac{r}{q}$.

Le taux de transmission décroit exponentiellement quand m croît.
$\Longrightarrow$ Les codes de Reed-Muller sont asymptotiquement mauvais.

Codes poinçonnés

- On va évaluer sur des points $P_1, \ldots, P_n$ de $\mathbb{F}_q^m$, avec $n < q^m$.
- Cela revient à considérer des codes de longueur $n < q^m$.
- Les paramètres n, k et d vont potentiellement décroitre.
- On espère que le taux $R = k/n$, et la distance minimale relative $\delta = d/n$ vont devenir bons.

On a donc la fonction d'évaluation :

$$\begin{array}{rccc} \mathrm{ev}: & \mathbb{F}_q[X_1, \ldots, X_m] & \to & \mathbb{F}_q^n \\ & f & \mapsto & (f(P_1), \ldots, f(P_n)) \end{array},$$

et le code associé est

$$C = \mathrm{ev}(L_r).$$

Problème : comment évaluer les paramètres ?

Comment poinçonner

On va prendre les points $P_1, \ldots, P_n \in \mathbb{F}_q^m$ de manière algébrique : sur une courbe algébrique.

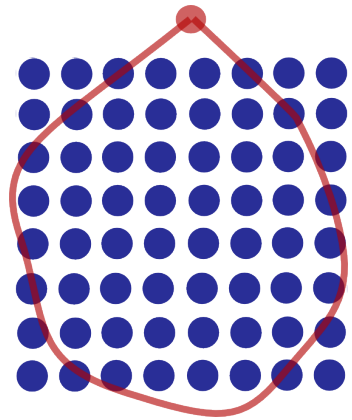


FIG. 3: Une courbe dans $\mathbb{F}_8^2$

On va considérer des courbes « en position spéciale » : une seule place centrée sur un unique point à l'infini, et la courbe est non singulière dans sa partie affine.

Comment poinçonner

On va prendre les points $P_1, \ldots, P_n \in \mathbb{F}_q^m$ de manière algébrique : sur une courbe algébrique.

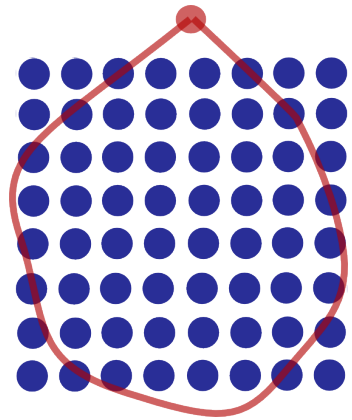


FIG. 3: Une courbe dans $\mathbb{F}_8^2$

On va considérer des courbes « en position spéciale » : une seule place centrée sur un unique point à l'infini, et la courbe est non singulière dans sa partie affine.
Toute courbe peut être mise en position spéciale (sans contrôle de la dimension de $\mathbb{A}^m$).

Simplifications

Proposition

Soit $\mathcal{C}$ une courbe affine dans $\mathbb{A}^m(\mathbb{F}_q)$, comme précédemment. Il existe $o_1, \ldots, o_m \in \mathbb{N}^\star$ tel que, pour tout monôme $m = x_1^{i_1} \cdots x_m^{i_m}$, l'ordre au pôle de m en Q est égal à

$$v_Q(x_1^{i_1} \ldots x_m^{i_m}) = -\left(o_1 i_1 + \cdots + o_m i_m\right).$$

On définit le degré pondéré $\mathrm{wdeg}(x_1^{i_1} \ldots x_m^{i_m}) = o_1 i_1 + \cdots + o_m i_m$.

Exemple (Courbe hermitienne)

Soit $\mathbb{F}_q = \mathbb{F}_{q_0^2}$, et $\mathcal{C}$ la courbe $x^{q_0+1} = y^{q_0} - y$.
Le point à l'infini est $(1 : 0 : 0)$.
On a $-v_Q(x) = q_0$ et $-v_Q(y) = q_0 + 1$.
L'ensemble

$$\left\{x^i y^j \middle| \quad 0 \leqslant i, \quad 0 \leqslant j \leqslant q_0\right\}$$

est une base de l'espace vectoriel $\mathbb{F}_q[\mathcal{C}]$.

Espaces L

Définition

Soit $\mathcal{C}$ une courbe affine en position spéciale, et Q la place à l'infini. L'espace associé à rQ, noté $L(rQ)$ est l'ensemble des fonctions $f \in \mathbb{F}_q(\mathcal{C})$ n'admettant qu'un unique pôle en Q, d'ordre au plus r.

On note $D = P_1 + \cdots + P_n$ (somme formelle).

Définition

Soit $\mathcal{C}$ une courbe comme précédemment, et Q la place à l'infini, et $D = P_1 + \cdots + P_n$, où $P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$.
Le code géométrique $C_L(D, rQ)$ est

$$\mathrm{ev}(L(rQ)).$$

Le code géométrique $C_\Omega(D, rQ)$ est le dual de $C_L(D, rQ)$.

Genre

Définition

Soit $\mathcal{C}$ une courbe comme précédemment, Q la place à l'infini, et wdeg *le degré pondéré induit par Q. Le semi-groupe de Weierstrass G_Q en Q est l'ensemble des poids des monômes de $\mathbb{F}_q[X_1, \ldots, X_m]$.*

C'est bien un semi-groupe, car si $u_1 = -v_Q(f_1)$ et $u_2 = -v_Q(f_2)$, alors $u_1 + u_1 = -v_Q(f_1 f_2)$.

Théorème

Pour un entier $u \in \mathbb{N}$ assez grand, $u + \mathbb{N} \subset G_Q$.

Définition

Soit $\mathcal{C}$ une courbe comme précédemment, et Q la place à l'infini. Le genre g de $\mathcal{C}$ est le cardinal de $\mathbb{N} \setminus G_Q$.

Courbes hermitiennes

Exemple

Soit la courbe hermitienne $\mathcal{C}$ définie sur $\mathbb{F}_{16}$ par $x^5 = y^4 + y$, on a $-v_{\mathcal{C}}(x) = 4$ et $-v_{\mathcal{C}}(y) = 5$.
Donc le semi-groupe de Weierstrass G_Q, qui est égal à $4\mathbb{N} + 5\mathbb{N}$, est

$$\{0, 4, 5, 8, 9, 10, 12, 13, 14, 15, 16, 17, \ldots\}$$

et il manque $1, 2, 3, 6, 7, 11$. Le genre de la courbe est donc 6.

De manière plus générale, la courbe hermitienne $X^{q_0+1} = Y^{q_0} + Y$ a q_0^3 points affines dans $\mathbb{F}_{q_0^2}$, et son genre est $q_0(q_0 - 1)/2$.

Note

Ces courbes admettent un autre modèle, d'équation $X^{q_0+1} + Y^{q_0+1} + Z^{q_0+1} = 0$.

« Théorème de Riemann-Roch »

Théorème (« Riemann-Roch »)

On a $\dim L(rQ) \geqslant r - g + 1$, *avec égalité si* $r > 2g - 2$.

Théorème

Une fonction sur la courbe a autant de zéros que de pôles (bien comptés). En particulier, une fonction $f \in L(rQ)$ *a plus* r *zéros.*

Proposition

Pour $r > 2g - 2$, *le code géométrique* $C_L(D, rQ)$ *est un code*

$$[n, k = r - g + 1, d \geqslant n - r]_q.$$

Les paramètres k et d de ces codes vérifient $d \geqslant n - k + 1 - g$.

« Théorème de Riemann-Roch »

Théorème (« Riemann-Roch »)

On a $\dim L(rQ) \geqslant r - g + 1$, *avec égalité si* $r > 2g - 2$.

Théorème

Une fonction sur la courbe a autant de zéros que de pôles (bien comptés). En particulier, une fonction $f \in L(rQ)$ *a plus* r *zéros.*

Proposition

Pour $r > 2g - 2$, *le code géométrique* $C_L(D, rQ)$ *est un code*

$$[n, k = r - g + 1, d \geqslant n - r]_q.$$

Les paramètres k et d de ces codes vérifient $d \geqslant n - k + 1 - g$.

Le défaut par rapport à la borne de Singleton est le genre g.

Exemple

- $\mathbb{F}_{256} = \mathbb{F}_{q_0^2} = \mathbb{F}_{16^2}$, la courbe hermitienne $x^{17} - y^{16} - y = 0$ a $q_0^3 = 2^{12} = 4096$ points affines.
- À comparer avec 256 points pour le code de Reed-Solomon.
- Le genre de la courbe est $\frac{q_0(q_0-1)}{2} = 120$.
- On obtient une famille de codes de dimension variable $[n = 4096, k = r + 1 - 120, d \geqslant 4096 - r]_{256}$ pour $r > 2g - 2$.

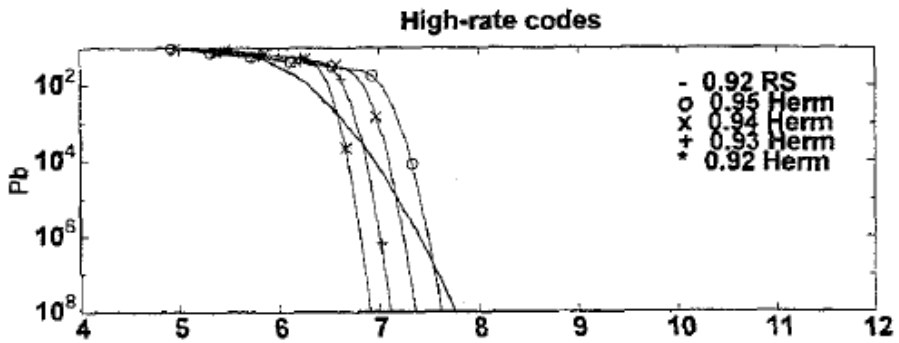


FIG. 4: B. E. Wahlen and J. Jimenez. Performance comparison of Hermitian and Reed-Solomon codes. In *MILCOM 97 Proceedings*, 1997

Hasse-Weil

On cherche donc des courbes avec beaucoup de points sur $\mathbb{F}_q$, et de genre contrôlé.

Théorème (Hasse-Weil)

Soit $\mathcal{C}$ une courbe définie sur $\mathbb{F}_q$, de genre g. Le nombre de points $N_q(\mathcal{C})$ sur $\mathbb{F}_q$ vérifie

$$|N_q(\mathcal{C}) - (q+1)| \leq 2g\sqrt{q}. \tag{1}$$

Pour les codes géométriques $C_L(D, rQ)$, on a les rapports suivants :

$$R = \frac{k}{n} = \frac{(r-g+1)}{n}, \quad \text{et} \quad \delta = \frac{d}{n} = \frac{n-r}{n} = 1 - \frac{r}{n}.$$

Donc

$$R \approx 1 - \delta - \frac{g}{n}.$$

La quantité asymptotique d'intérêt est $A(q) = \limsup_{g\to\infty} \frac{n}{g}$.

Tsfasman-Vladut-Zink

Théorème (Tsfasman-Vladut-Zink)

Si q est un carré, $A(q) = \limsup_{g\to\infty} \frac{n}{g} = \sqrt{q} - 1$.

> *M. A. Tsfasman, S. G. Vlăduţ, and Th. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound.* Math. Nachr, *1982*

Théorème

Soit q un carré, et R et δ tels que

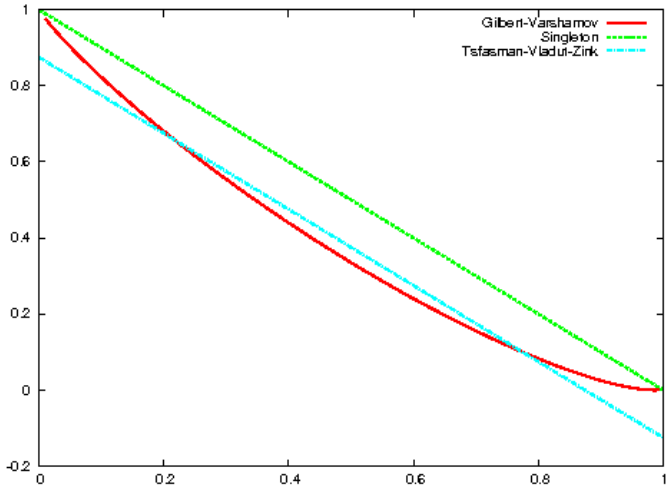
$$R + \delta = 1 - \frac{1}{\sqrt{q} - 1}.$$

Il existe une famille de codes $[n_i, k_i, d_i]_q$ telle que $\frac{k_i}{n_i} \longrightarrow R$ et $\frac{d_i}{n_i} \longrightarrow \delta$.

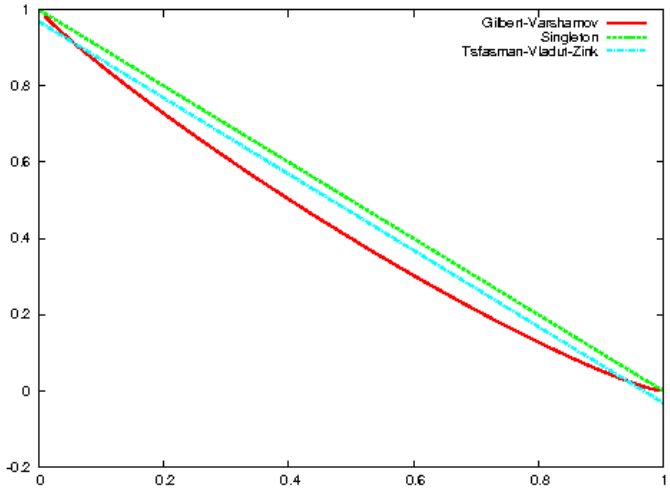
Meilleur que Varshamov-Gilbert, i.e. que les codes aléatoires.

Graphiques

Si $q = \square \geqslant 49$, on a des codes meilleurs que les codes aléatoires !



$q = 64$,



$q = 1024$.

Tours de Garcia et Stichenoth

- La complexité initale donnée par Tsfasman et Vladut était de $O(n^{30})$ (sic !).
- Garcia et Stichenoth ont proposé une série de corps de fonctions $F_0, F_1, \ldots, F_i, \ldots$ avec

$$F_0 = \mathbb{F}_{q_0^2}(x_0), \quad F_i = F_{i-1}[x_i], \quad \text{où } x_i^{q_0} + x_i = \frac{x_{i-1}^{q_0}}{x_{i-1}^{q_0-1} + 1}$$

telle que $\limsup \frac{n_i}{g_i} = q_0 - 1$.

A. Garcia and H. Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. Journal of Number Theory, *1996*

Construction effective en $O\left((n \log n)^3\right)$.

Plan

Position du problème

Le code $\mathrm{RS}(n, k)$ est le dual de $\mathrm{RS}(n, n-k)$, qui est

$$\mathrm{RS}(n, n-k) = \{\big(f(\alpha^0, \ldots, f(\alpha^{n-1}))\big)\,;\ \deg f < n-k\}$$

Le code $\mathrm{RS}(n, k)$ est fixé une fois pour toutes.

- L'inconnue est $c \in C$, le mot tranmis ;
- le « mot reçu » $y \in \mathbb{F}_q^n$ est connu, tel que $y = c + e$, où e est l'erreur, inconnue (additive) ;
- le but est de retrouver e à partir de y, de manière efficace ;
- si $\mathrm{w}(e) \leq t = \lfloor \frac{d-1}{2} \rfloor$, il y a unicité de e.

Décodage par syndrome

Puisque $\mathrm{RS}(n,k)$ est le dual de $\mathrm{RS}(n,n-k)$, tout mot de code $c \in \mathrm{RS}(n,k)$ vérifie

$$c(\alpha^0) = \cdots = c(\alpha^{n-k-1}) = 0.$$

- Si $y \in \mathbb{F}_q^n$ est le « mot reçu », tel que $y = c + e$, on a

$$y(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i), \quad i \in \{0, \ldots, n-k-1\}.$$

Définition

Les

$$S_i = y(\alpha^i), \quad i \in \{0, \ldots, n-k-1\}$$

sont les syndromes de l'erreur (relativement au code).

$\Longleftarrow$ Ils ne dépendent pas du mot transmis.

$\Longleftarrow$ Ils caractérisent uniquement l'erreur, quand $\mathrm{w}(e) \leq t$.

Connexion avec la transformée de Fourier discrète

Soit $\alpha \in \mathbb{F}_q$ une racine n-ième de l'unité.
Soit $c \in \mathbb{F}_q^n$, $c = c(x) = \sum_{i=0}^{n-1} c_i x^i$.
La transformée de Fourier discrète de c est

$$C = (C_0, \ldots, C_{n-1}), \quad C_i = c(\alpha^i), \quad i \in \{0, \ldots, n-1\}$$

Le code est l'ensemble des mots tels que

$$c(\alpha^0) = \cdots = c(\alpha^{n-k-1}) = 0.$$

Pour $y = c + e$, on a

$$y(\alpha^i) = e(\alpha^i), \quad i \in \{0, \ldots, n-k-1\}.$$

On dit aussi que

- les $S_i = y(\alpha^i) = e(\alpha^i)$, $i \in \{0, \ldots, n-k-1\}$ sont les syndromes connus,
- les $S_i = e(\alpha^i)$, $i \in \{n-k, \ldots, n-1\}$ sont les syndromes inconnus.

Dérivation de l'« équation clé »

Définition

Soit $e = (e_0, \ldots, e_{n-1})$ *un erreur de poids* w*. Le support de l'erreur est*

$$I = \{i \in \{0, \ldots, n-1\} | e_i \neq 0\}$$

et on définit

$$\{X_1, \ldots X_w\} = \{\alpha^i | i \in I\}$$

qui sont les localisateurs d'erreur de e*.*

On considéra le problème résolu quand on a trouvé les localisateurs.

Dérivation de l'« équation clé »

Définition

Soit $e = (e_0, \ldots, e_{n-1})$ *un erreur de poids* w*. Le support de l'erreur est*

$$I = \{i \in \{0, \ldots, n-1\} | e_i \neq 0\}$$

et on définit

$$\{X_1, \ldots X_w\} = \{\alpha^i | i \in I\}$$

qui sont les localisateurs d'erreur de e*.*

On considéra le problème résolu quand on a trouvé les localisateurs. Logarithmes discrets, mais longueur petite $\Longrightarrow$ recherche exhaustive.

Polynôme localisateur

Définition

Soit e une erreur de localisateurs $X_1, \ldots, X_w$, le polynôme localisateur de e est

$$\sigma = \prod_{i=1}^{w} (1 - X_i x).$$

Le polynôme évaluateur de e est

$$\omega = \sum_{i \in I} e_i \prod_{j \in \{1,\ldots,w\} \setminus \{i\}} (1 - X_j x).$$

On considéra le problème résolu quand on a trouvé le polynôme localisateur.

$\Longleftarrow$ n est petit, on peut faire une recherche exhaustive sur les racines n-ièmes de l'unité.

Dérivation

On calcule :

$$\begin{aligned}
\frac{\omega(x)}{\sigma(x)} &= \sum_{i\in I} \frac{e_{j_i}}{1-\alpha^{j_i}x} \\
&= \sum_{i\in I} e_{j_i} \sum_{l=0}^{\infty} \left(\alpha^{j_i}x\right)^l \\
&= \sum_{l=0}^{\infty} x^l \sum_{i\in I} e_{j_i} \left(\alpha^l\right)^{j_i} \\
&= \sum_{l=0}^{\infty} x^l e(\alpha^l) \\
&= \sum_{l=0}^{\infty} S_l x^l.
\end{aligned}$$

« L'équation clé »

On a

$$\frac{\omega(x)}{\sigma(x)} = \sum_{l=0}^{\infty} S_l x^l,$$

mais le décodeur ne connait que les S_l, $l \in \{0, \ldots, n-k-1\}$ (les « syndromes »).
On ne peut donc qu'écrire :

$$\begin{aligned} \frac{\omega(x)}{\sigma(x)} &= \sum_{l=0}^{n-k-1} S_l x^l \bmod x^{n-k}, \\ &= S(x) \bmod x^{n-k} \end{aligned}$$

avec $\deg \sigma \leq t$, et $\deg \omega \leq t-1$, et $S(x) = \sum_{l=0}^{n-k-1} S_l x^l$.

Donc $\frac{\omega}{\sigma}$ est un $(t+1, t)$ approximant de Padé de S.
$\Longrightarrow$ Algorithe d'Euclide étendu.

Algorithme d'Euclide étendu

L'« équation clé » est

$$\frac{\omega}{\sigma} = S \bmod X^{n-k}.$$

En déroulant l'algorithme d'Euclide étendu entre x^{n-k} et S, on maintient l'invariant :

$$s_i x^{n-k} + t_i S = r_i,$$

où (r_i) est la suite des restes de degrés décroissants, et (s_i), (t_i) sont les suites des coefficients de Bezout.

Proposition

Pour le premier i tel que $\deg r_i \leq t-1$, *on a* $\sigma = t_i$ *et* $\omega = r_i$.

Algorithme d'Euclide étendu

L'« équation clé » est

$$\frac{\omega}{\sigma} = S \bmod X^{n-k}.$$

En déroulant l'algorithme d'Euclide étendu entre x^{n-k} et S, on maintient l'invariant :

$$s_i x^{n-k} + t_i S = r_i,$$

où (r_i) est la suite des restes de degrés décroissants, et (s_i), (t_i) sont les suites des coefficients de Bezout.

Proposition

Pour le premier i tel que $\deg r_i \leq t-1$, *on a* $\sigma = t_i$ *et* $\omega = r_i$.

$\Longleftarrow$ unicité de la solution

Algorithme d'Euclide étendu

L'« équation clé » est

$$\frac{\omega}{\sigma} = S \bmod X^{n-k}.$$

En déroulant l'algorithme d'Euclide étendu entre x^{n-k} et S, on maintient l'invariant :

$$s_i x^{n-k} + t_i S = r_i,$$

où (r_i) est la suite des restes de degrés décroissants, et (s_i), (t_i) sont les suites des coefficients de Bezout.

Proposition

Pour le premier i tel que $\deg r_i \leq t-1$, *on a* $\sigma = t_i$ *et* $\omega = r_i$.

$\Longleftarrow$ unicité de la solution $\Longleftarrow$ unicité de l'erreur

Algorithme d'Euclide étendu

L'« équation clé » est

$$\frac{\omega}{\sigma} = S \bmod X^{n-k}.$$

En déroulant l'algorithme d'Euclide étendu entre x^{n-k} et S, on maintient l'invariant :

$$s_i x^{n-k} + t_i S = r_i,$$

où (r_i) est la suite des restes de degrés décroissants, et (s_i), (t_i) sont les suites des coefficients de Bezout.

Proposition

Pour le premier i tel que $\deg r_i \leq t-1$, *on a* $\sigma = t_i$ *et* $\omega = r_i$.

$\Longleftarrow$ unicité de la solution $\Longleftarrow$ unicité de l'erreur $\Longleftarrow$ $t = \lfloor \frac{d-1}{2} \rfloor$.

Euclide ou Berlekamp-Massey ?

L'équation clé

$$\frac{\omega}{\sigma} = S \bmod x^{n-k}.$$

entraîne (pour $t = \lfloor \frac{d-1}{2} \rfloor = \lfloor \frac{n-k}{2} \rfloor$)

$$\sum_{i=0}^{t} \sigma_{j-i} S_i = 0, \quad j = t \ldots 2t.$$

Comme $\sigma_0 = 1$ par construction, on a la relation de récurrence :

$$S_j = -\sum_{i=1}^{t} \sigma_i S_{j-i}, \quad j = t \ldots 2t. \tag{2}$$

Le polynôme $\sigma = \sigma(x) = \sum \sigma_i x^i$ définit une relation de récurrence satisfaite par la suite $S_1, \ldots, S_{2t}$. $\Longrightarrow$ Berlekamp-Massey.

Euclide ou Berlekamp-Massey

L'algorithme de Berlekamp-Massey produit la plus petite relation de récurrence linéaire satisfaite par une suite.

- J. Dornstetter. On the equivalence between Berlekamp's and Euclid's algorithms. *IEEE Trans. on Inf. Theory*, 1987
- A. E. Heydtmann and J. M. Jensen. On the equivalence of the Berlekamp-Massey and the Euclidean algorithms for decoding. *IEEE Trans. on Inf. Theory*, 2000
- Maria Bras-Amorós and Michael O'Sullivan. From the euclidean algorithm for solving a key equation for dual Reed–solomon codes to the Berlekamp–Massey algorithm. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 2009

Euclide ou Berlekamp-Massey

L'algorithme de Berlekamp-Massey produit la plus petite relation de récurrence linéaire satisfaite par une suite.

- J. Dornstetter. On the equivalence between Berlekamp's and Euclid's algorithms. *IEEE Trans. on Inf. Theory*, 1987
- A. E. Heydtmann and J. M. Jensen. On the equivalence of the Berlekamp-Massey and the Euclidean algorithms for decoding. *IEEE Trans. on Inf. Theory*, 2000
- Maria Bras-Amorós and Michael O'Sullivan. From the euclidean algorithm for solving a key equation for dual Reed–solomon codes to the Berlekamp–Massey algorithm. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 2009

On décodera des codes « multivariés » où les anneaux ne sont plus euclidiens. Mais il est possible de généraliser Berlekamp-Massey.

Algorithme de Berlekamp-Massey

À chaque étape j, un polynôme σ est maintenu tel que

$$S_k = -\sum_{i=1}^{\deg \sigma} \sigma_i S_{k-i}, \quad k = \deg \sigma \ldots j.$$

Pour passer à l'étape $j+1$, il faut prendre en compte S_{j+1}.
À l'étape $j+1$, le polynôme courant σ est testé, en calculant

$$\overline{S}_{j+1} = -\sum_{i=1}^{\deg \sigma} \sigma_i S_{j-i}, \quad \text{la « prédiction ».}$$

Si $\overline{S}_{j+1} = S_{j+1}$, alors σ est conservé. Sinon, on a une discrépance

$$\delta = S_{j+1} - \overline{S}_{j+1}.$$

Et on met à jour σ de la manière suivante : $\sigma \leftarrow \sigma - \delta\delta_0^{-1} x^l \sigma_0$, où σ_0 est le polynôme qui a produit la discrépance $\delta_0 \neq 0$, dans une étape précécente.

Entrée : $S_1, \ldots, S_j$
σ valide jusqu'à l'étape $j-1$
σ_0 ayant échoué avant l'étape $j-1$.
δ_0 la discrépance à l'endroit où σ_0 a échoué.
l : le nombre de fois où σ n'a pas changé de degré.
L le degré de σ.

Sortie : σ^+, σ_0^+, δ_0^+, l^+, L^+.

$\overline{S_j} \leftarrow S_j + \sum \sigma_i S_{j-i}$
if $\overline{S_j} = S_j$ **then**
    $l = l + 1$
else
    if $2L \geqslant j$ **then**
        $\sigma^+ \leftarrow \sigma - \delta\delta_0^{-1} x^l \sigma_0$, et $l \leftarrow l + 1$
    else
        $\sigma^+ \leftarrow \sigma - \delta\delta_0^{-1} x^l \sigma_0$, $\sigma_0^+ \leftarrow \sigma$
        $L^+ \leftarrow j - L$
    end if
end if

Algorithme de Berlekamp-Massey

- Complexité $O(t^2)$;
- facile à programmer, forme matricielle, forme « dans le domaine de la transformée » ;
- plus rapide pour les BCH (gain d'un facteur 2) ;
- il existe une version rapide (si convolutions rapides) ;
- bien maîtrisé en matériel ;

Algorithme de Berlekamp-Massey

- Complexité $O(t^2)$;
- facile à programmer, forme matricielle, forme « dans le domaine de la transformée » ;
- plus rapide pour les BCH (gain d'un facteur 2) ;
- il existe une version rapide (si convolutions rapides) ;
- bien maîtrisé en matériel ;
- se généralise pour les codes géométriques.

Décodage des codes géomeriques C_Ω : le code

- Courbes en position spéciale où Q est la place à l'infini.
- $C_\Omega(D, rQ) = C_L(D, rQ)^\perp$, avec

$$C_L(D, rQ) = \{(f(P_1), \ldots, f(P_n)) \, ; f \in L(rQ)\}$$

- Soit $x_1, \ldots, x_m$ les fonctions coordonnées de $\mathbb{A}^m(\mathbb{F}_q)$, et $-o_1, \ldots, -o_m$ leurs ordres en Q. On a l'ordre monomial :

$$\text{wdeg}(\mathbf{x^s}) = \text{wdeg}(x_1^{s_1} \cdots x_m^{s_m}) = o_1 s_1 + \cdots + o_m s_m.$$

- Une famille génératrice de $C_L(D, P)$ est donc

$$\{(x^{\mathbf{s}}(P_1), \ldots, x^{\mathbf{s}}(P_n)); \ \text{wdeg}\, \mathbf{x^s} \leq r\}.$$

- Les mots de code $c = (c_1, \ldots, c_n)$ sont ceux qui vérifient :

$$\sum_{i=1}^{n} c_i x^{\mathbf{s}}(P_i) = 0; \quad \text{pour tout } \mathbf{s} \text{ tel que } \text{wdeg}(\mathbf{s}) \leqslant r.$$

Syndromes

- Dans la situation du décodage, on a $y = c + e$, où c est le mot de code, e est l'erreur et y est le mot reçu.
- Pour $\mathbf{s}$ tel que $\mathrm{wdeg}(\mathbf{s}) \leq r$:

$$\sum_{i=1}^{n} y_i \mathbf{x}^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} c_i \mathbf{x}^{\mathbf{s}}(P_i) + \sum_{i=1}^{n} e_i x^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i).$$

- La connaissance de y (le mot reçu), donne la connaissance des syndromes (qui ne dépendent que de l'erreur e) :

$$E_{\mathbf{s}} = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \mathrm{wdeg}\, \mathbf{s} \leq r.$$

- Si la distance minimale est $2t + 1$, alors pour un ensemble de syndromes $E_{\mathbf{s}}$, $\mathrm{wdeg}(\mathbf{s}) \leqslant r$, il existe au plus une erreur de poids t admettant ces syndromes.

Transformée généralisée

Définition

Soit $\mathcal{P} = \{P_1, \ldots, P_n\} \subset \mathbb{F}_q^m$, *la transformée généralisée sur* $\mathbb{F}_q^n$ *est la fonction*

$$\begin{array}{rccc} \mathcal{F} : & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^{\mathbb{N}^m} \\ & e & \mapsto & E = (E_{\mathbf{s}})_{\mathbf{s} \in \mathbb{N}^m} \end{array}$$

où

$$E_{\mathbf{s}} = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \mathbf{s} \in \mathbb{N}^m.$$

Le tableau $E = (E_{\mathbf{s}})$ de la transformée est multi-dimensionnel et infini.
Dans le cas $\mathbb{F}_q$, le tableau E est cyclique dans chaque dimension.

Si $m = 1$, et $(P_1, \ldots, P_n) = (\alpha^0, \alpha^1, \ldots, \alpha^{n-1})$, on retrouve la transformée de Fourier classique.

Syndromes connus et inconnus

Dans le cas d'une erreur e, les syndromes connus sont

$$E_{\mathbf{s}} = \sum_{i=1}^{n} y_i \mathbf{x}^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \operatorname{wdeg} \mathbf{s} \leqslant r$$

et les autres

$$E_{\mathbf{s}} = \sum_{i=1}^{n} y_i \mathbf{x}^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \operatorname{wdeg} \mathbf{s} > r$$

sont les syndromes inconnus.

Syndromes connus et inconnus

Dans le cas d'une erreur e, les syndromes connus sont

$$E_{\mathbf{s}} = \sum_{i=1}^{n} y_i \mathbf{x}^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \operatorname{wdeg} \mathbf{s} \leqslant r$$

et les autres

$$E_{\mathbf{s}} = \sum_{i=1}^{n} y_i \mathbf{x}^{\mathbf{s}}(P_i) = \sum_{i=1}^{n} e_i \mathbf{x}^{\mathbf{s}}(P_i), \quad \operatorname{wdeg} \mathbf{s} > r$$

sont les syndromes inconnus.

On ne connait donc qu'une partie de la transformée généralisée.

Idéal localisateur d'erreurs

Définition

Soit

$$f = f(\mathbf{x}) = \sum f_{\mathbf{s}} x^{\mathbf{s}} \in \mathbb{F}[\mathbf{x}],$$

on dit que le tableau m-dimensionnel infini $E = (E_{\mathbf{s}})_{\mathbf{s}\in\mathbb{N}^m}$ *vérifie la relation de récurrence* f *si*

$$\sum_{\mathbf{s}} f_{\mathbf{s}} E_{\mathbf{s}+\mathbf{r}} = 0, \quad \text{pour tout } \mathbf{r} \geqslant 0. \tag{3}$$

On dit aussi que la relation de récurrence représentée par le polynôme $f \in \mathbb{F}[x_1, \ldots, x_m]$ *est valide pour le tableau* E*.*

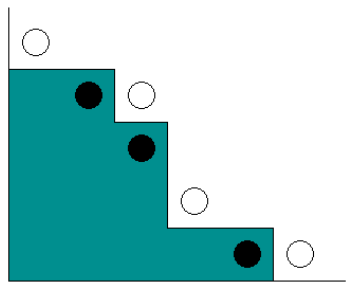
Proposition

Soit e *une erreur de transformée* $(E_{\mathbf{s}})_{\mathbf{s}\in\mathbb{N}^m}$*. L'ensemble des relations de récurrence valides pour le tableau* E *définit un idéal,* $V(E)$*. Ses zéros sont les points* P_i *tels que* $e_i \neq 0$*.*

Algorithme de Berlekamp-Massey-Sakata

> *S. Sakata. Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array.* J. Symb. Comput., *1988*

Un ensemble delta est complètement déterminé par ses coins extérieurs de la manière suivante $\Delta = \{\mathbf{r} : \forall \mathbf{u} \in \mathrm{Ext}\,\Delta,\ \mathbf{u} \not\leqslant \mathbf{r}\}$.



L'algorithme de Berlekamp-Massey-Sakata essaye de trouver les polynômes minimaux valides pour le tableau des syndrômes (connus et inconnus).

$\Longrightarrow$ base de Gröbner de l'idéal localisateur des positions d'erreur.

Polynômes valides et invalides

Soit E un tableau m-dimensionnel, et f valide pour le tableau E. Pour un ordre monomial donné, et un indice du tableau $\mathbf{u}$ tel que $\mathrm{lead}(f)|\mathbf{u}$, la relation

$$\sum_{\mathbf{s}} f_{\mathbf{s}} E_{\mathbf{s}+\mathbf{r}} = 0, \quad \text{pour tout } \mathbf{r} \geqslant 0$$

se réécrit en

$$E_{\mathbf{u}} = \frac{-1}{\mathrm{lc}(f)} \sum_{\mathbf{p}<\mathbf{u}} f_{\mathrm{lead}(f)-\mathbf{u}+\mathbf{p}} E_{\mathbf{p}}.$$

Définition

Un polynôme f est invalide à la position $\mathbf{u}$ si $\mathrm{lead}(f)|\mathbf{u}$ et si

$$E_{\mathbf{u}} \neq \frac{-1}{\mathrm{lc}(f)} \sum_{\mathbf{p}<\mathbf{u}} f_{\mathrm{lead}(f)-\mathbf{u}+\mathbf{p}} E_{\mathbf{p}}.$$

Sinon, f est valide à la position $\mathbf{u}$.

Progression dans le tableau

Note

Un polynôme tel que $\mathrm{lead}(f) \nmid \mathbf{u}$ *est automatiquement valide à la position* $\mathbf{u}$.

Proposition

Si f *est valide à la position* $\mathbf{u}$, *alors* $x^{\mathbf{p}} f$ *est valide à la position* $\mathbf{u}$.

Définition

Un polynôme f *est valide jusqu'à la position* $\mathbf{u}$ *s'il est valide à toutes les positions* $\mathbf{r} \leqslant \mathbf{u}$. *On note* $V_{\mathbf{u}}(E)$ *l'ensemble de polynômes valides jusqu'à la position* $\mathbf{u}$.

Soit $\mathbf{u}$ un indice du tableau, et $\mathbf{u}^+$ l'indice suivant pour l'ordre monomial. On a alors les inclusions suivantes :

$$\begin{array}{ccccccc} \mathbf{0} & < & \mathbf{u} & < & \mathbf{u}^+ & < & \infty \\ \mathbb{F}_q[\mathbf{x}] & \supseteq & V_{\mathbf{u}}(E) & \supseteq & V_{\mathbf{u}^+}(E) & \supseteq & V(E) \\ \emptyset & \subseteq & \Delta(V_{\mathbf{u}}(E)) & \subseteq & \Delta(V_{\mathbf{u}} + (E)) & \subseteq & \Delta(V(E)) \end{array}$$

Prédictions

Définition

Soit $\mathbf{u}$ *un indice du tableau* E*, et* f *valide pour toutes les positions* $\mathbf{r} \leqslant \mathbf{u}$*. La prédiction de* f *à l'indice suivant* $\mathbf{u}^+$ *est :*

$$P_{\mathbf{u}^+}(f) = \frac{-1}{\mathrm{lc}(f)} \sum_{\mathbf{p}<\mathbf{u}^+} f_{\mathrm{lead}(f)-\mathbf{u}^++\mathbf{p}} E_{\mathbf{p}}.$$

Proposition (Prédictions)

Soit f *et* g *valides jusqu'à* $\mathbf{u}$*, et tels que* $\mathrm{lead}(f) + \mathrm{lead}(g)$ *divise* $\mathbf{u}^+$*. Alors les prédictions de* f *et de* g *en* $\mathbf{u}^+$ *sont les mêmes :*

$$P_{\mathbf{u}^+}(f) = P_{\mathbf{u}^+}(g).$$

Portée (span)

Définition

Soit g valide jusqu'à la position $\mathbf{u}$ et invalide en $\mathbf{u}^+$. La portée (span) de g est

$$\mathrm{span}(g) = \mathbf{u}^+ - \mathrm{lead}(g).$$

Théorème

Soit $g \notin V_{\mathbf{u}^+}(E)$. Alors $\mathrm{span}(g) \in \Delta(V_{\mathbf{u}^+}(E))$.

Définition

Soit $g \in \mathbb{F}[\mathbf{x}] \setminus V(E)$. On dit que g est un témoin de $\mathrm{span}(g)$.
Soit $\mathcal{G} \in \mathbb{F}[\mathbf{x}] \setminus V(E)$, et Δ un ensemble delta.
On dit que $\mathcal{G}$ est un ensemble témoin de Δ, si $\mathcal{G}$ contient un témoin de chaque sommet intérieur de Δ.

Itération de $\mathbf{u}$ à $\mathbf{u}^+$

Pour l'indice courant $\mathbf{u}$, on maintient deux ensembles $\mathcal{F}$ et $\mathcal{G}$:

- $\mathcal{F}$: un ensemble de polynômes valides jusqu'à la position $\mathbf{u}$: il y en un par élément extérieur de l'ensemble Δ de $V_{\mathbf{u}}(E)$;
- $\mathcal{G}$: un ensemble de polynômes témoins de l'ensemble Δ de $V_{\mathbf{u}}(E)$.

Le principe est de combiner des polynômes valides jusqu'à $\mathbf{u}$, donc issus de $\mathcal{F}$, à des polynômes précédemment invalides, donc issus de $\mathcal{G}$, pour passer de $\mathbf{u}$ à $\mathbf{u}^+$.

Construction de $\mathcal{G}^+$

1. On teste les prédictions $P_{\mathbf{u}^+}(f)$ des polynômes de $\mathcal{F}$.
2. Ceux qui échouent sont ajoutés à $\mathcal{G}^+$.
3. Les portées des polynômes de $\mathcal{G}+$ forment un ensemble témoin de l'ensemble Δ de $V_{\mathbf{u}^+}(E)$.

Itération de $\mathbf{u}$ à $\mathbf{u}^+$

Pour chaque sommet extérieur à $\mathbf{s} \in \Delta(V_{\mathbf{u}^+}(E)))$, on va contruire un polynôme valide en $\mathbf{u}^+$. Il y a trois cas :

1. soit il y a un ancien polynôme de terme de tête $\mathbf{s}$ valide en $\mathbf{u}$ qui est toujours valide en $\mathbf{u}^+$: on le recycle ;
2. soit $\mathbf{s}$ ne divise pas $\mathbf{u}^+$, alors on « décale » un polynôme f valide en $\mathbf{u}$ pour que son terme de tête soit égal à $\mathbf{s}$;
3. soit $\mathbf{s}$ divise $\mathbf{u}^+$: on met à jour un polynôme $f \in \mathcal{F}$ en le combinant avec un polynôme de $\mathcal{G}$ pour avoir une prédiction juste en $\mathbf{u}^+$.

Itération de $\mathbf{u}$ à $\mathbf{u}^+$

Pour chaque sommet extérieur à $\mathbf{s} \in \Delta(V_{\mathbf{u}^+}(E)))$, on va contruire un polynôme valide en $\mathbf{u}^+$. Il y a trois cas :

1. soit il y a un ancien polynôme de terme de tête $\mathbf{s}$ valide en $\mathbf{u}$ qui est toujours valide en $\mathbf{u}^+$: on le recycle ;
2. soit $\mathbf{s}$ ne divise pas $\mathbf{u}^+$, alors on « décale » un polynôme f valide en $\mathbf{u}$ pour que son terme de tête soit égal à $\mathbf{s}$;
   (il existe un tel f)

3. soit $\mathbf{s}$ divise $\mathbf{u}^+$: on met à jour un polynôme $f \in \mathcal{F}$ en le combinant avec un polynôme de $\mathcal{G}$ pour avoir une prédiction juste en $\mathbf{u}^+$.

Itération de $\mathbf{u}$ à $\mathbf{u}^+$

Pour chaque sommet extérieur à $\mathbf{s} \in \Delta(V_{\mathbf{u}^+}(E)))$, on va contruire un polynôme valide en $\mathbf{u}^+$. Il y a trois cas :

1. soit il y a un ancien polynôme de terme de tête $\mathbf{s}$ valide en $\mathbf{u}$ qui est toujours valide en $\mathbf{u}^+$: on le recycle ;
2. soit $\mathbf{s}$ ne divise pas $\mathbf{u}^+$, alors on « décale » un polynôme f valide en $\mathbf{u}$ pour que son terme de tête soit égal à $\mathbf{s}$;
   (un polynôme dont le terme de tête ne divise pas $\mathbf{u}^+$ est valide en $\mathbf{u}^+$)
3. soit $\mathbf{s}$ divise $\mathbf{u}^+$: on met à jour un polynôme $f \in \mathcal{F}$ en le combinant avec un polynôme de $\mathcal{G}$ pour avoir une prédiction juste en $\mathbf{u}^+$.

Itération de $\mathbf{u}$ à $\mathbf{u}^+$

Pour chaque sommet extérieur à $\mathbf{s} \in \Delta(V_{\mathbf{u}^+}(E)))$, on va contruire un polynôme valide en $\mathbf{u}^+$. Il y a trois cas :

1. soit il y a un ancien polynôme de terme de tête $\mathbf{s}$ valide en $\mathbf{u}$ qui est toujours valide en $\mathbf{u}^+$: on le recycle ;
2. soit $\mathbf{s}$ ne divise pas $\mathbf{u}^+$, alors on « décale » un polynôme f valide en $\mathbf{u}$ pour que son terme de tête soit égal à $\mathbf{s}$;
   (un polynôme dont le terme de tête ne divise pas $\mathbf{u}^+$ est valide en $\mathbf{u}^+$)
3. soit $\mathbf{s}$ divise $\mathbf{u}^+$: on met à jour un polynôme $f \in \mathcal{F}$ en le combinant avec un polynôme de $\mathcal{G}$ pour avoir une prédiction juste en $\mathbf{u}^+$. (Sorte de S-polynôme)

Entrée : E, $\mathbf{u}$, $\mathbf{u}^+$, $\mathcal{F}$ ensemble minimal de $V_{\mathbf{u}}(E)$, $\mathcal{G}$ ensemble témoin de $V_{\mathbf{u}}(E)$

Sortie : $\mathcal{F}^+$, $\mathcal{G}^+$.

Corps de la boucle :

$\mathcal{N} \leftarrow$ les éléments de $\mathcal{F}$ invalides en $\mathbf{u}^+$
$\mathcal{G}^+ \leftarrow \mathcal{G} \cup \mathcal{N}$; $\Delta^+ \leftarrow \mathrm{span}(\mathcal{G}^+)$
for $\mathbf{s} \in \mathrm{Ext}\,\Delta^+$ **do**
    if $\exists f \in \mathcal{F} \setminus \mathcal{N}$ tel que $\mathrm{lead}(f) = \mathbf{s}$ **then**
        $h^{\mathbf{s}} \leftarrow f$
    else if $\mathbf{s} \nmid \mathbf{u}^+$ **then**
        Trouver $f \in \mathcal{N}$ avec $\mathrm{lead}(f) | \mathbf{s}$; $h^{\mathbf{s}} \leftarrow x^{\mathbf{s}-\mathrm{lead}(f)} f$
    else
        Trouver $f \in \mathcal{N}$, tel que $\mathrm{lead}(f) \mid \mathbf{s}$
        Trouver $g \in \mathcal{G}$, tel que $(\mathbf{u}^+ - \mathbf{s}) \mid \mathrm{span}(g)$
        $h^{\mathbf{s}} \leftarrow S_{\mathbf{s},\mathbf{u}^+}(f, g)$
    end if
end for
$\mathcal{F}^+ = \{h^{\mathbf{s}}; \mathbf{s} \in \mathrm{Ext}\,\Delta^+\}$

Procédure de vote

En ne prenant en compte que les « syndromes connus », la procédure précédente décode $\frac{d_G-1-g}{2}$ erreurs.

L'idée est d'essayer d'étendre le nombre de syndromes connus, en utilisant les calculs déjà faits.

Procédure de vote

En ne prenant en compte que les « syndromes connus », la procédure précédente décode $\frac{d_G-1-g}{2}$ erreurs.

L'idée est d'essayer d'étendre le nombre de syndromes connus, en utilisant les calculs déjà faits.
Redondance ?

Procédure de vote

En ne prenant en compte que les « syndromes connus », la procédure précédente décode $\frac{d_G - 1 - g}{2}$ erreurs.

L'idée est d'essayer d'étendre le nombre de syndromes connus, en utilisant les calculs déjà faits.

Redondance ?

Supposons que le syndrome $E_{\mathbf{u}+}$ soit inconnu.
Les polynômes valides f jusqu'à la position $E_{\mathbf{u}}$ font des prédictions $P_{\mathbf{u}^+}(f)$.

Proposition

La prédiction juste a la majorité relative parmi toutes les prédictions.

> *G. L. Feng and T. R. N. Rao. Decoding algebraic-geometric codes up to the designed minimum distance.* IEEE Trans. on Inf. Theory, *1993*

Encapsulation

Entrée : $E_{\mathbf{s}}$, pour $\operatorname{wdeg} \mathbf{s} \leq \mathbf{r}$, le tableau des syndrômes connus.
Initalisations $\mathbf{u} = 0$, $\mathcal{F} = \{1\}$, $\mathcal{G} = \emptyset$
Répéter

1: **if** Le syndrôme $E_{\mathbf{u}^+}$ est inconnu **then**
2: Utiliser la procédure de vote pour obtenir $E_{\mathbf{u}^+}$.
3: **end if**
4: Utiliser l'étape élémentaire Berlekamp-Massey-Sakata pour trouver $\mathcal{F}^+$, $\mathcal{G}^+$
5: $\mathcal{F} \leftarrow \mathcal{F}^+$
6: $\mathcal{G} \leftarrow \mathcal{G}^+$
7: $\mathbf{u} \leftarrow \mathbf{u}^+$

Jusqu'à ce que suffisamment de syndrômes soient déterminés.

Encapsulation

Entrée : $E_{\mathbf{s}}$, pour wdeg $\mathbf{s} \leq \mathbf{r}$, le tableau des syndrômes connus.
Initalisations $\mathbf{u} = 0$, $\mathcal{F} = \{1\}$, $\mathcal{G} = \emptyset$
Répéter

1: **if** Le syndrôme $E_{\mathbf{u}^+}$ est inconnu **then**
2: Utiliser la procédure de vote pour obtenir $E_{\mathbf{u}^+}$.
3: **end if**
4: Utiliser l'étape élémentaire Berlekamp-Massey-Sakata pour trouver $\mathcal{F}^+$, $\mathcal{G}^+$
5: $\mathcal{F} \leftarrow \mathcal{F}^+$
6: $\mathcal{G} \leftarrow \mathcal{G}^+$
7: $\mathbf{u} \leftarrow \mathbf{u}^+$

Jusqu'à ce que suffisamment de syndrômes soient déterminés.

Démontré quand $t \leq \frac{d'-1}{2}$ (distance de Feng et Rao).

Encapsulation

Entrée : $E_{\mathbf{s}}$, pour wdeg $\mathbf{s} \leq \mathbf{r}$, le tableau des syndrômes connus.
Initalisations $\mathbf{u} = 0$, $\mathcal{F} = \{1\}$, $\mathcal{G} = \emptyset$
Répéter

1: **if** Le syndrôme $E_{\mathbf{u}^+}$ est inconnu **then**
2: Utiliser la procédure de vote pour obtenir $E_{\mathbf{u}^+}$.
3: **end if**
4: Utiliser l'étape élémentaire Berlekamp-Massey-Sakata pour trouver $\mathcal{F}^+$, $\mathcal{G}^+$
5: $\mathcal{F} \leftarrow \mathcal{F}^+$
6: $\mathcal{G} \leftarrow \mathcal{G}^+$
7: $\mathbf{u} \leftarrow \mathbf{u}^+$

Jusqu'à ce que suffisamment de syndrômes soient déterminés.

Démontré quand $t \leq \frac{d'-1}{2}$ (distance de Feng et Rao).
Explosif quand on augmentre t.

Deçodage par syndrome : bilan

- Cas des Reed-Solomon : bien éprouvé, implantation matérielle, ubiquité ;
- cas des codes géométriques : bien étudié, implantation matérielle,
- Berlekamp-Massey-Sakata a une complexité de $O(T^2)$, pour un tableau de taille T, « version rapide »d'après Sakata ;
- version pipelinée de Berlekamp-Massey-Sakata ;
- pas de déploiement ;

Deçodage par syndrome : bilan

- Cas des Reed-Solomon : bien éprouvé, implantation matérielle, ubiquité ;
- cas des codes géométriques : bien étudié, implantation matérielle,
- Berlekamp-Massey-Sakata a une complexité de $O(T^2)$, pour un tableau de taille T, « version rapide »d'après Sakata ;
- version pipelinée de Berlekamp-Massey-Sakata ;
- pas de déploiement ; trop compliqué ?

Deçodage par syndrome : bilan

- Cas des Reed-Solomon : bien éprouvé, implantation matérielle, ubiquité ;
- cas des codes géométriques : bien étudié, implantation matérielle,
- Berlekamp-Massey-Sakata a une complexité de $O(T^2)$, pour un tableau de taille T, « version rapide »d'après Sakata ;
- version pipelinée de Berlekamp-Massey-Sakata ;
- pas de déploiement ; trop compliqué ?
- sauf pour le canal à effacement (Shokrollahi).

Deçodage par syndrome : bilan

- Cas des Reed-Solomon : bien éprouvé, implantation matérielle, ubiquité ;
- cas des codes géométriques : bien étudié, implantation matérielle,
- Berlekamp-Massey-Sakata a une complexité de $O(T^2)$, pour un tableau de taille T, « version rapide »d'après Sakata ;
- version pipelinée de Berlekamp-Massey-Sakata ;
- pas de déploiement ; trop compliqué ?
- sauf pour le canal à effacement (Shokrollahi).

Pas de possibilité de décoder plus d'erreurs que le nombre donné par les théorèmes.

Plan

Position du problème

- Soit $\mathbb{F}_q$ le corps fini de cardinal q, $n \leq q$, et $x_1, \ldots, x_n \in \mathbb{F}_q$ distincts. Nous avons la fonction d'évaluation associée à $(x_1, \ldots, x_n)$, qui est

$$\begin{array}{rccc} \mathrm{ev} : & \mathbb{F}_q[x] & \to & \mathbb{F}_q^n \\ & f & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

- Le code de Reed-Solomon, de longueur n, de support $(x_1, \ldots, x_n)$, de dimension k est

$$\{\mathrm{ev}\, f, f \in \mathbb{F}_q[x], \deg f < k\}.$$

- La distance minimale est $d = n - k + 1$, et ce code peut corriger $t = \lfloor \frac{n-k}{2} \rfloor$ erreurs.

- Soit $c = \mathrm{ev}\, f$ le mot transmis, et $y = (y_1, \ldots, y_n)$ le mot reçu. On considére le problème du décodage résolu quand f aura été retrouvé à partir de y.

Position du problème

- Soit $\mathbb{F}_q$ le corps fini de cardinal q, $n \leq q$, et $x_1, \ldots, x_n \in \mathbb{F}_q$ distincts. Nous avons la fonction d'évaluation associée à $(x_1, \ldots, x_n)$, qui est

$$\begin{array}{rccc} \mathrm{ev} : & \mathbb{F}_q[x] & \to & \mathbb{F}_q^n \\ & f & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

- Le code de Reed-Solomon, de longueur n, de support $(x_1, \ldots, x_n)$, de dimension k est

$$\{\mathrm{ev}\, f, f \in \mathbb{F}_q[x], \deg f < k\}.$$

- La distance minimale est $d = n - k + 1$, et ce code peut corriger $t = \lfloor \frac{n-k}{2} \rfloor$ erreurs. (Pour simplifier on suppose que $n - k$ est pair, et donc que $t = \frac{n-k}{2}$)
- Soit $c = \mathrm{ev}\, f$ le mot transmis, et $y = (y_1, \ldots, y_n)$ le mot reçu. On considére le problème du décodage résolu quand f aura été retrouvé à partir de y.

Interpolation de bas degré

Données

- $x_1, \ldots, x_n$, distincts (constants dans le cadre du codage)
- $(y_1, \ldots, y_n) \in \mathbb{F}_q^n$ (le mot reçu dans le cadre du codage)
- $k < n$ un degré.

Question

Trouver un polynôme de degré inférieur à k, tel que

$$|\{i| \quad f(x_i) \neq y_i\}| \leq \frac{n-k}{2}.$$

Exemple : $k = 2$, trouver la droite qui passe par au moins $n - \frac{n-2}{2}$ points.

« Interpolation de Lagrange avec erreurs »

Interpolation de bas degré

Données

- $x_1, \ldots, x_n$, distincts (constants dans le cadre du codage)
- $(y_1, \ldots, y_n) \in \mathbb{F}_q^n$ (le mot reçu dans le cadre du codage)
- $k < n$ un degré.

Question

Trouver un polynôme de degré inférieur à k, tel que

$$|\{i| \quad f(x_i) \neq y_i\}| \leq \frac{n-k}{2}.$$

Exemple : $k = 2$, trouver la droite qui passe par au moins $n - \frac{n-2}{2}$ points.
« Approximation pour la norme L^0 »

Berlekamp-Welch : mise en équation

Définition

Pour un mot y reçu, et $c = \mathrm{ev}\, f$ le mot transmis, le polynôme localisateur est $E = E(X) = \prod_{i;f(x_i)\neq y_i}(X - x_i)$.

Observation

Pour tout $i \in \{1, \ldots, n\}$, de deux choses l'une : ou bien

$$f(x_i) = y_i,$$

ou bien

$$E(x_i) = 0.$$

Ce qui se traduit par $f(x_i)E(x_i) = y_iE(x_i)$, $i \in \{1, \ldots, n\}$.

- Système à n équations,
- Les inconnues sont les coefficients de f et les coefficients de E.
- Non linéaire, degré deux.

Berlekamp-Welch : linéarisation

On a donc le système :

$$f(x_i)E(x_i) = y_i E(x_i), i \in \{1, \ldots, n\}.$$

avec $\deg E \leq t$ et $\deg f < k$.

Système linéaire

On introduit le polynôme $F = f \cdot E$.
On obtient donc le système d'équations linéaires.

$$F(x_i) = y_i E(x_i), \quad i \in \{1, \ldots, n\}.$$

avec $\deg F < t + k$, $\deg E \leq t$.

- Nombre d'équations : n ;
- Nombre d'inconnues :

$$t + k + t + 1 = 2t + k + 1 = k + d = n + 1.$$

Berlekamp-Welch : linéarisation

On a donc le système :

$$f(x_i)E(x_i) = y_i E(x_i), i \in \{1, \ldots, n\}.$$

avec $\deg E \leq t$ et $\deg f < k$.

Système linéaire

On introduit le polynôme $F = f \cdot E$.
On obtient donc le système d'équations linéaires.

$$F(x_i) = y_i E(x_i), \quad i \in \{1, \ldots, n\}.$$

avec $\deg F < t + k$, $\deg E \leq t$.

- Nombre d'équations : n ;
- Nombre d'inconnues :

$$t + k + t + 1 = 2t + k + 1 = k + d = n + 1. \quad \text{Miracle !}$$

Berlekamp-Welch : correction

Proposition

Pour tout solution E, F du système, si le nombre d'erreurs est inférieur ou égal à t, alors le polynôme $f = \frac{F}{E} \in \mathbb{F}_q[x]$, f est la bonne solution.

Démonstration.

Soit $E_1, F_1 \in \mathbb{F}_q[X]$ et $E_2, F_2 \in \mathbb{F}_q[X]$ deux couples de solutions de ce système. Comme on a

$$F_1(x_i) = y_i E_1(x_i) \text{ et } F_2(x_i) = y_i E_2(x_i), \quad i \in \{1, \ldots, n\},$$

On déduit $E_2(x_i)F_1(x_i) - E_1(x_i)F_2(x_i) = 0 \quad i \in \{1, \ldots, n\}$. Or

$$\deg(E_2F_1 - E_1F_2) < t + (k + t) = n$$

Il a n racines, il est identiquement nul. Donc $E_2F_1 - E_1F_2 = 0$. □

Berlekamp-Welch : correction

Proposition

Pour tout solution E, F du système, si le nombre d'erreurs est inférieur ou égal à t, alors le polynôme $f = \frac{F}{E} \in \mathbb{F}_q[x]$, f est la bonne solution.

Démonstration.

Soit $E_1, F_1 \in \mathbb{F}_q[X]$ et $E_2, F_2 \in \mathbb{F}_q[X]$ deux couples de solutions de ce système. Comme on a

$$F_1(x_i) = y_i E_1(x_i) \text{ et } F_2(x_i) = y_i E_2(x_i), \quad i \in \{1, \ldots, n\},$$

On déduit $E_2(x_i)F_1(x_i) - E_1(x_i)F_2(x_i) = 0 \quad i \in \{1, \ldots, n\}$. Or

$$\deg(E_2F_1 - E_1F_2) < t + (k + t) = n \quad \text{(Re-Miracle !)}$$

Il a n racines, il est identiquement nul. Donc $E_2F_1 - E_1F_2 = 0$. □

Algorithme de Berlekamp-Welch

Constantes :

q, $n < q$, k, t, tels que $n - k$ est pair, et $t = \frac{n-k}{2}$.

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\operatorname{ev} f, y) \leq t$.

Interpolation : Trouver $E, F \in F_q[x]$ tels que

1. $F(x_i) - y_i E(x_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\deg F < t + k$;
3. $\deg E \leq t$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\operatorname{ev} f, y) > t$, déclarer un échec de décodage.

Échec de décodage

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Codes géométriques C_L

Le code

- $\mathcal{C}$ de genre g, en position spéciale, admettant une unique place à l'infini Q ;
- Soit $\{P_1, \ldots, P_n\} \in \mathcal{C}(\mathbb{F}_q)$ des points affines disctints ;
- le code est

$$C_L(\mathcal{P}, rQ) = \{\text{ev}\, f = (f(P_1), \ldots, f(P_n));\ f \in L(rQ)\}\,.$$

Le problème

Soit $c = \text{ev}\, f$ le mot transmis, et soit $y = (y_1, \ldots, y_n)$ le mot reçu.
Il faut retrouver $f \in L(rQ)$, telle que

$$|\{i|\ f(P_i) \neq y_i\}| \leq \tau.$$

Changement de notation : $\tau \neq t = \lfloor \frac{d-1}{2} \rfloor$.

Observation

- On cherche une « fonction localisatrice d'erreur », $E \in \mathbb{F}_q[\mathcal{C}]$, telle que $E(P_i) = 0$, uniquement pour $i \in I$.
- Pour borner son ordre au pôle, il suffit de considérer le diviseur $D = (\tau + g)Q - \sum_{i \in I} P_i$.
  Par le théorème de Riemann, on a

$$\dim L(D) \geqslant \deg D - g + 1 = 1.$$

  Donc $-v_Q(E) \leq \tau + g$.

Mise en équation

Pour tout $i \in \{1, \ldots, n\}$ on a : $f(P_i) = y_i$, ou $E(P_i) = 0$, ce qui se traduit par

$$f(P_i)E(P_i) = y_i E(P_i), \quad i \in \{1, \ldots, n\}.$$

Linéarisation

On a

$$f(P_i)E(P_i) = y_i E(P_i), \quad i \in \{1, \ldots, n\}.$$

avec $f \in L(rQ)$, et $E \in L(\tau + g)$.
On cherche $Q(Y) = F + YE \in \mathbb{F}_q[\mathcal{C}][Y]$ tel que :

- $F(P_i) - y_i E(P_i) = 0$, $i \in \{1, \ldots, n\}$,
- $F \in L((\tau + g + r)Q)$
- $E \in L(\tau + g)Q)$

Proposition

Pour toute solution E, F du système, si le nombre d'erreurs est inférieur à $\tau < \lfloor \frac{d-1-g}{2} \rfloor$, alors $f = -\frac{E}{F}$ est la « bonne solution ».

Linéarisation

On a

$$f(P_i)E(P_i) = y_i E(P_i), \quad i \in \{1, \ldots, n\}.$$

avec $f \in L(rQ)$, et $E \in L(\tau + g)$.
On cherche $Q(Y) = F + YE \in \mathbb{F}_q[\mathcal{C}][Y]$ tel que :

- $F(P_i) - y_i E(P_i) = 0$, $i \in \{1, \ldots, n\}$,
- $F \in L((\tau + g + r)Q) \iff \text{wdeg}(F) \leqslant \tau + g + r$
- $E \in L(\tau + g)Q) \iff \text{wdeg}(E) \leqslant \tau + g$.

Proposition

Pour toute solution E, F du système, si le nombre d'erreurs est inférieur à $\tau < \lfloor \frac{d-1-g}{2} \rfloor$, alors $f = -\frac{E}{F}$ est la « bonne solution ».

Algorithme pour les codes C_L

Constantes :

q, n, r, τ.

$\mathcal{C}$ une courbe affine de genre g, en « position spéciale ».

$(P_1, \ldots, P_n) \in \mathcal{C}(\mathbb{F}_q)$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : $f \in L(rQ)$, telle que $d(\operatorname{ev} f, y) \leq \tau$

Interpolation : Trouver $E, F \in \mathbb{F}_q[\mathcal{C}]$ tels que

1. $F(P_i) - y_i E(P_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\operatorname{wdeg}(F) < \tau + g + r$;
3. $\operatorname{wdeg}(E) \leq \tau + g$

Recherche de racines : Retourner une bonne représentation de $f = -\frac{F}{E}$.

Si $d(\operatorname{ev} f, y) > t$ ou $f \notin L(rQ)$, déclarer un échec.

Décodage en liste

Principe

Si on relâche l'idée de l'uncitié de la solution, en admettant de retourner une liste de mots de codes. on peut se permettre d'augmenter le nombre d'erreurs à $\tau > t = \lfloor \frac{d-1}{2} \rfloor$.

Jusqu'où ?

Si τ devient trop grand, on risque de retourner un trop grand nombre de mots.

$\Longrightarrow$ voire exponentiel en n, si $k = R \cdot n$.

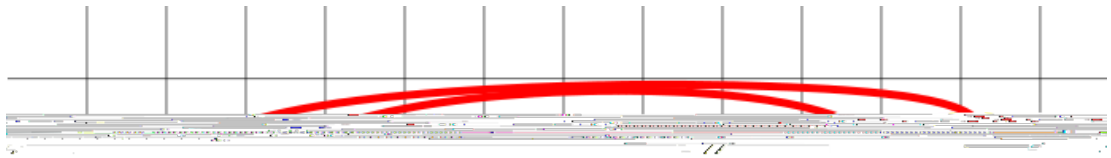
Théorisé, non réalisé jusqu'à Sudan

P. Elias. Error-correcting codes for list decoding. *IEEE Trans. on Inf. Theory*, 1991

« list-of-L decoding » $\Longrightarrow$ Borne de Hamming, de Varshamov-Gilbert généralisées.

Idée de Sudan

On construit $Q(X, Y)$ qui passe par les points (x_i, y_i).



Alors tous les $f(X)$ cherchés sont tels que $Q(x, f(X)) = 0$.

Algorithme de Berlekamp-Welch

Constantes :

q, $n < q$, k, et $t = \frac{n-k}{2}$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\operatorname{ev} f, y) \leq t$

Interpolation : Trouver $E, F \in F_q[x]$ tels que

1. $F(x_i) - y_i E(x_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\deg F < t + k$;
3. $\deg E \leq t$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\operatorname{ev} f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $E, F \in F_q[x]$ tels que

1. $F(x_i) - y_i E(x_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\deg F < t + k$;
3. $\deg E \leq t$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $F(x_i) - y_i E(x_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\deg F < t + k$;
3. $\deg E \leq t$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\deg F < t + k$;
3. $\deg E \leq t$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $Q = \sum_j Q_j(x) y^j$
3. $\deg Q_i < n - \tau - (k-1)i$

Recherche de racines : Retourner $f = -\frac{F}{E}$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $Q = \sum_j Q_j(x) y^j$
3. $\deg Q_i < n - \tau - (k-1)i$

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Si $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > t$, déclarer un échec de décodage.

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $Q = \sum_j Q_j(x) y^j$
3. $\deg Q_i < n - \tau - (k-1)i$

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Échec

L'échec de décodage n'est pas une erreur de l'algorithme : le mot reçu est simplement trop « loin ».

Preuve

Proposition

Soit y le mot reçu, et $Q \neq 0 \in \mathbb{F}_q[x, y]$ tel que

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\mathrm{wdeg}_{1,k-1} Q < n - \tau$.

Si $d(\mathrm{ev}\, f, y) \leq \tau$, et $\deg f < k$, alors $Q(x, f(x)) = 0$.

Démonstration.

- $\deg f < k$, $\mathrm{wdeg}_{1,k-1} Q < n - \tau \Longrightarrow \deg Q(X, f(X)) < n - \tau$.
- $\forall i$, $Q(x_i, y_i) = 0$. Donc $f(x_i) = y_i$, $\Longrightarrow Q(x_i, f(x_i)) = 0$.
- Comme $d(\mathrm{ev}\, f, y) \leq \tau$, on a $f(x_i) = y_i$ pour au moins $n - \tau$ éléments x_i.
- $Q(x, f(x))$ a au moins $n - \tau$ zéros. $\Longrightarrow Q(x, f(x)) = 0$.

Analyse du nombre d'erreurs corrigibles

Dernier argument logique : il faut que, pour tout mot reçu, $Q(x, y) \neq 0$ existe tel que

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\mathrm{wdeg}_{1,k-1} Q < n - \tau$.

Analyse simple

Il suffit d'avoir strictement plus d'inconnues que d'équations.
Nombre d'inconnues

$$\sum_{i=0}^{l} (1 + \deg Q_i) = \sum_{i=0}^{l} (n - \tau) - (k-1)i > \frac{(n-\tau)^2}{2(k-1)}.$$

En écrivant donc $\frac{(n-\tau)^2}{2(k-1)} > n$, on extrait $\tau < n - \sqrt{2(k-1)n}$.

M. Sudan. Maximum likelihood decoding of Reed-Solomon codes. In *FOCS 1996*

Asymptotique

$$\tau < n - \sqrt{2(k-1)n} \implies \frac{\tau}{n} < 1 - \sqrt{1-2R}$$

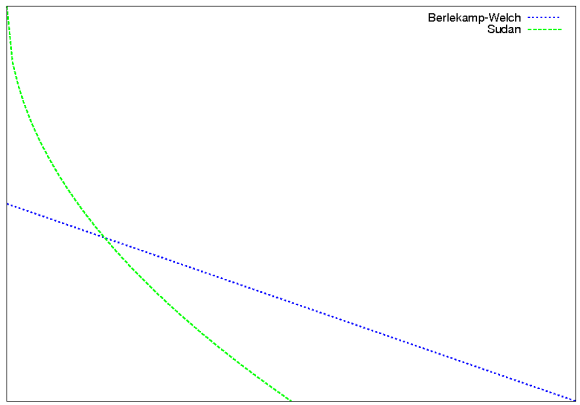


FIG. 5: Rayon $\frac{\tau}{n}$ de décodage, en fonction de $R = k/n$.

Asymptotique

$$\tau < n - \sqrt{2(k-1)n} \implies \frac{\tau}{n} < 1 - \sqrt{1-2R}$$

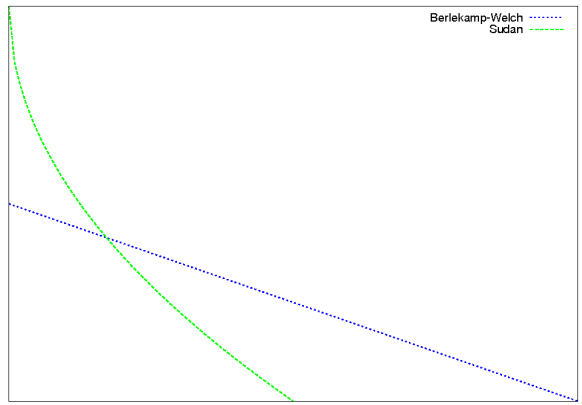


FIG. 5: Rayon $\frac{\tau}{n}$ de décodage, en fonction de $R = k/n$.

Matrice d'interpolation : block Vandermonde $\implies$ complexité $O(ln^2)$.

Asymptotique

$$\tau < n - \sqrt{2(k-1)n} \implies \frac{\tau}{n} < 1 - \sqrt{1-2R}$$

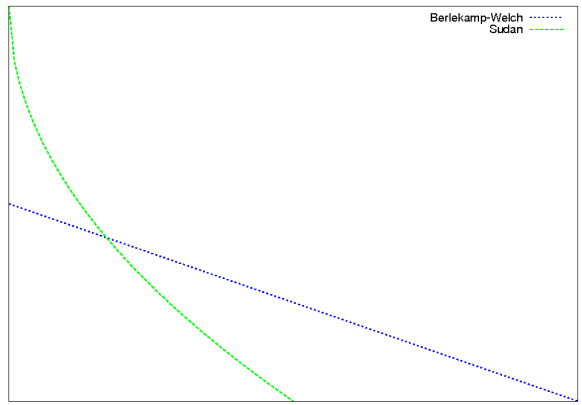


FIG. 5: Rayon $\frac{\tau}{n}$ de décodage, en fonction de $R = k/n$.

Matrice d'interpolation : block Vandermonde $\implies$ complexité $O(ln^2)$.

« Asymptotique » mais la taille de l'alphabet croît.

Asymptotique

$$\tau < n - \sqrt{2(k-1)n} \implies \frac{\tau}{n} < 1 - \sqrt{1-2R}$$

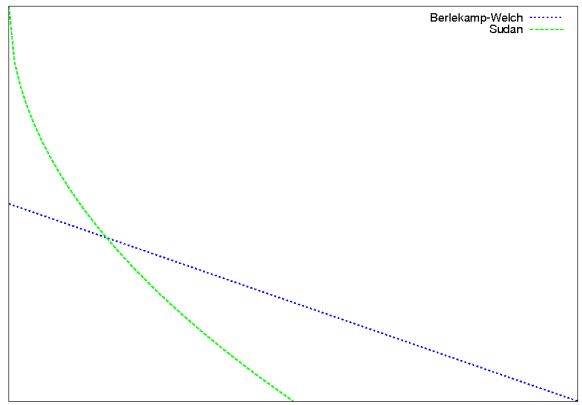


FIG. 5: Rayon $\frac{\tau}{n}$ de décodage, en fonction de $R = k/n$.

Matrice d'interpolation : block Vandermonde $\implies$ complexité $O(ln^2)$.

« Asymptotique » mais la taille de l'alphabet croît.

$\implies$ codes géométriques.

Sudan pour les codes de Reed-Solomon

Constantes :

n, k, et τ le rayon de correction.

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Tous les $f \in \mathbb{F}_q[X]$, $\deg f < k$, tels que $d(\operatorname{ev} f, y) \leq \tau$.

Interpolation : Trouver un polynôme $Q \neq 0 \in \mathbb{F}_q[X][Y]$ tels que

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\operatorname{wdeg}_{1,k-1} Q < n - \tau$.

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\operatorname{ev} f, y) > \tau$.

Si la liste est vide ; échec du décodage

Sudan pour les codes C_L à un point

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,

$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Tous les $f \in \mathbb{F}_q[X]$, $\deg f < k$, tels que $d(\mathrm{ev}\, f, y) \leq \tau$.

Interpolation : Trouver un polynôme $Q \neq 0 \in \mathbb{F}_q[X][Y]$ tels que

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\mathrm{wdeg}_{1,k-1} Q < n - \tau$.

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Sudan pour les codes C_L à un point

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,

$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Tous les $f \in L(rQ)$, tels que $d(\operatorname{ev} f, y) < \tau$.

Interpolation : Trouver un polynôme $Q \neq 0 \in \mathbb{F}_q[X][Y]$ tels que

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $\operatorname{wdeg}_{1,k-1} Q < n - \tau$.

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\operatorname{ev} f, y) > \tau$.

Si la liste est vide ; échec du décodage

Sudan pour les codes C_L à un point

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,

$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Tous les $f \in L(rQ)$, tels que $d(\operatorname{ev} f, y) < \tau$.

Interpolation : Trouver $Q = \sum_{i=0}^{l} Q_i Y^i \neq 0 \in \mathbb{F}_q[C][Y]$ tel que

1. $Q(y_i)(P_i) = 0$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((n - \tau - 1 - ri)Q)$;

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\operatorname{ev} f, y) > \tau$.

Si la liste est vide ; échec du décodage

Sudan pour les codes C_L à un point

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,

$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Tous les $f \in L(rQ)$, tels que $d(\mathrm{ev}\, f, y) < \tau$.

Interpolation : Trouver $Q = \sum_{i=0}^{l} Q_i Y^i \neq 0 \in \mathbb{F}_q[C][Y]$ tel que

1. $Q(y_i)(P_i) = 0$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((n - \tau - 1 - ri)Q)$;

Recherche de racines : Retourner les f tels que $Q(f) = 0$.

Retirer les f tels que $f \notin L(rQ)$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Preuve

Proposition

Soit $f \in L(rQ)$, tel que $d(\mathrm{ev}\, f, y) \leq \tau$, alors $Q(f) = 0$.

Démonstration.

- Par construction du polynôme, $Q(f) \in L((n - \tau - 1)Q)$.
- Comme $d(\mathrm{ev}\, f, y) \leq \tau$, on a $f(P_i) = y_i$ en au moins $\mu \geqslant n - \tau$ points $P_{i_1}, \ldots, P_{i_\mu}$.
- Donc

$$Q(f) \in L((n - \tau - 1)\, Q - (P_{i_1} + \cdots + P_{i_\mu})),$$

- Or $\mu \geqslant n - \tau$, donc le degré du diviseur $(n - \tau - 1)Q - (P_{i_1} + \cdots + P_{i_\mu})$ est négatif.
- L'espace associé est de dimension nulle : $Q(f) = 0$.

Analyse du nombre d'erreurs corrigibles

Dernier argument logique : il faut que $Q \neq 0$ existe tel que

1. $Q(y_i)(P_i) = 0$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((n - \tau - 1 - ri)Q)$;

Il suffit d'avoir moins de contraintes linéaires que la dimension autorisée pour l'espace des Q_i.

$$\sum_{i=0}^{l} \dim L((n-\tau-1-ri)Q) \geqslant \sum_{i=0}^{l} (n-\tau) - ri - g > \frac{(n-\tau-g)^2}{2r}.$$

En écrivant $\frac{(n-\tau-g)^2}{2r} > n$, on extrait

$$\tau < n - \sqrt{2rn} - g.$$

En terme de la dimension, on obtient

$$\tau < n - \sqrt{2(k+g-1)n} - g.$$

Analyse du nombre d'erreurs corrigibles

Dernier argument logique : il faut que $Q \neq 0$ existe tel que

1. $Q(y_i)(P_i) = 0$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((n - \tau - 1 - ri)Q)$;

Il suffit d'avoir moins de contraintes linéaires que la dimension autorisée pour l'espace des Q_i.

$$\sum_{i=0}^{l} \dim L((n-\tau-1-ri)Q) \geqslant \sum_{i=0}^{l} (n-\tau) - ri - g > \frac{(n-\tau-g)^2}{2r}.$$

En écrivant $\frac{(n-\tau-g)^2}{2r} > n$, on extrait

$$\tau < n - \sqrt{2rn} - g.$$

En terme de la dimension, on obtient

$$\tau < n - \sqrt{2(k+g-1)n} - g.$$

À comparer à $n - \sqrt{2(k-1)n}$.

Promesses théoriques

Proposition (Borne de Johnson)

Soit C un code de longueur n et de distance minimale d. Soit $y \in \mathbb{F}_q^n$, et $B(y, \tau) \cap C$ l'ensemble des mots de codes à distance τ de y. Alors, pour tout y,

$$|B(y, \tau) \cap C| \leq \frac{n(d - \tau)}{\tau^2 - 2n\tau + dn},$$

pourvu que le dénominateur soit positif, ce qui est vérifié pour $\tau < n - \sqrt{n(n-d)}$ (rayon de Johnson)

On est garanti d'avoir une taille de liste polynomiale $O(n^2)$, quand τ est inférieur au rayon de décodage.

Condition nécessaire, pour les codes quelconques

Il existe des codes (non explicites) tels que la taille de liste est exponentielle dès qu'on dépasse le rayon de Johnson.

Cas de codes de Reed-Solomon

Théorème (Guruswami-Vardy, 2005)

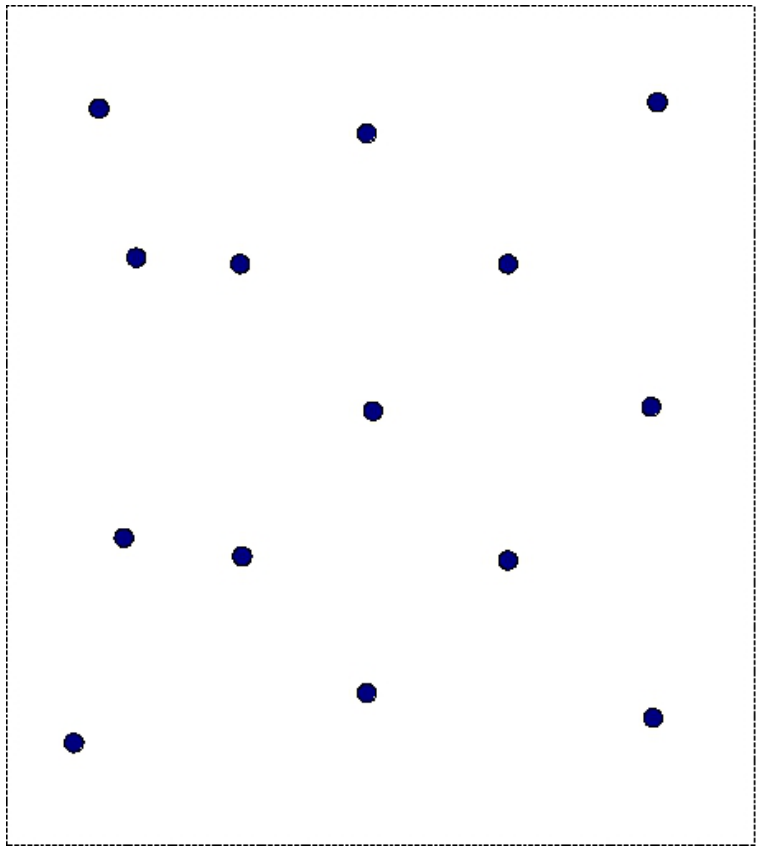
Le problème de décision associé au décodage des codes de Reed-Solomon est NP-complet.

- Dans la réduction, la taille de l'alphabet doit croître vite.
- Théorème équivalent pour le genre 1, même croissance de l'alphabet.
- Il existe des bornes sur τ, telles qu'au dessus de ces bornes, il existe des mots y admettant un nombre non polynomial de mots de code à distance τ.

  Eli B. Sasson, S. Kopparty, and J. Radhakrishnan. Subspace polynomials and list decoding of Reed-Solomon codes. In FOCS '06

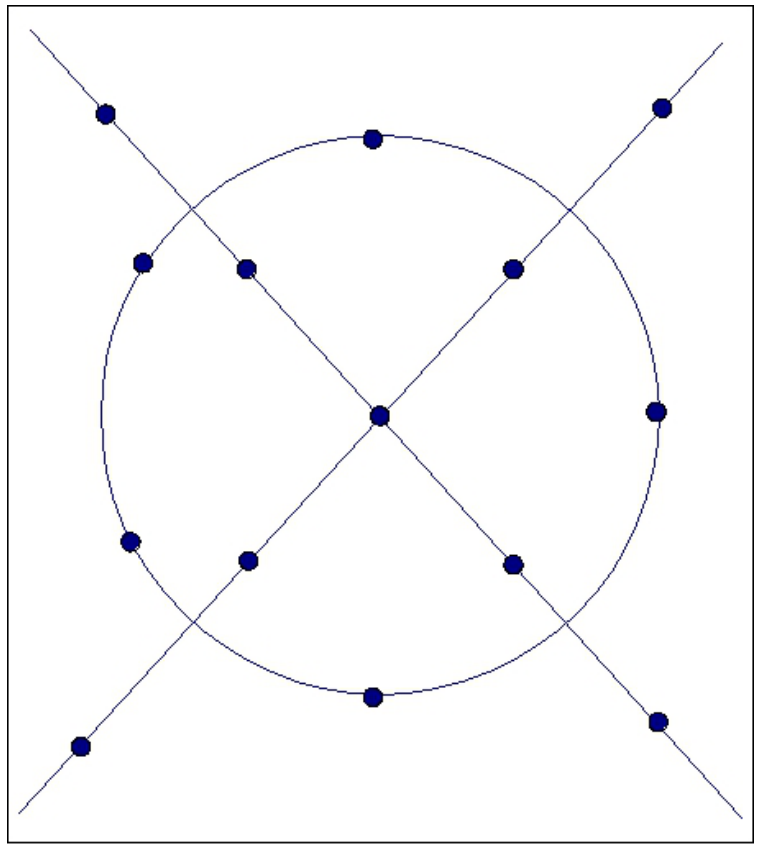
- Mais il reste un écart entre la borne de Johnson et ces bornes.

Exemples (forcés)



Quelles sont les droites $y = a_i x + b_i$ passant par au moins 5 points ?

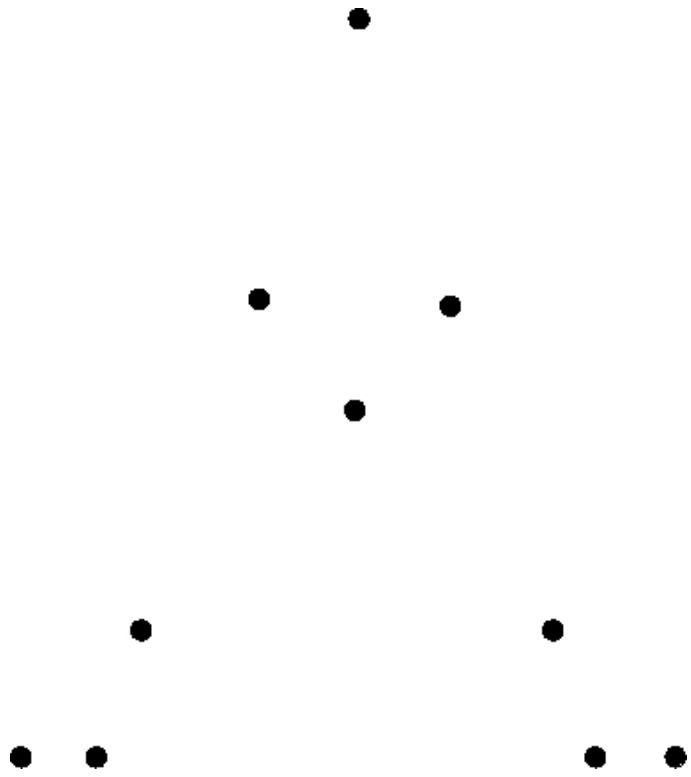
Exemples (forcés)



On a le cercle qui est une scorie pour le décodage.

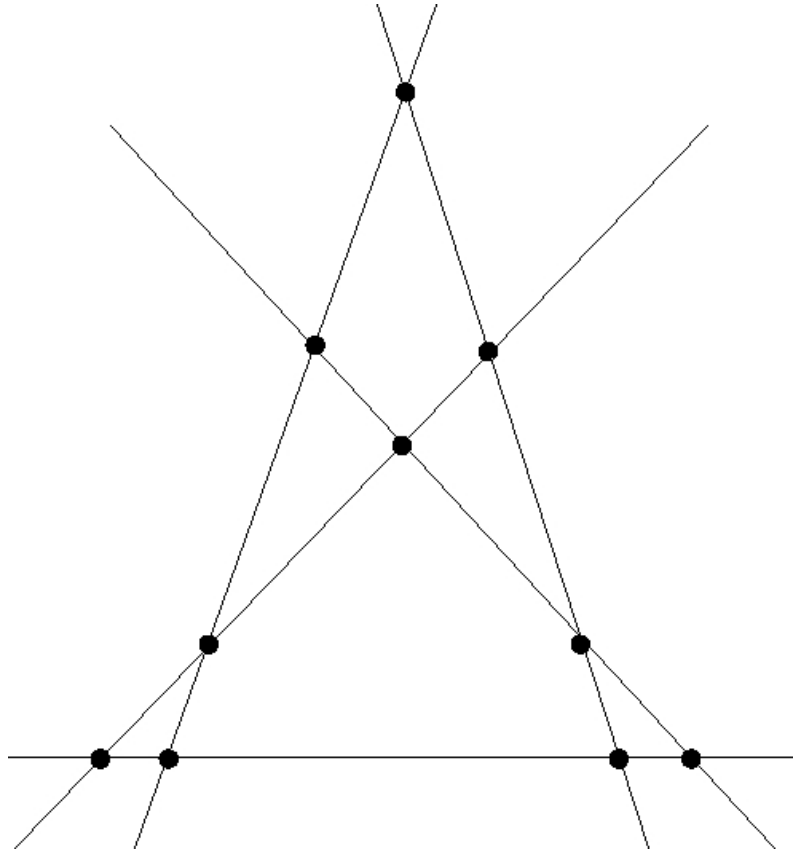
$$Q(x,y) = C(x,y)\prod_{i=1}^{2}(y - a_i x - b_i)$$

Exemples (forcés)



Quelles sont les droites $y = a_i x + b_i$ passant par au moins 4 points ?

Exemples (forcés)



Par chaque point considéré passe deux droites. Si, pour chaque droite $f_i = a_i x + b_i$, on a $Q(x, f(x)) = 0$, alors

$$\text{mult}(Q, (x_j, y_j)) = 2, \quad j \in \{1, \ldots, 10\}$$

Algorithme de Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq t$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $Q(x_i, y_i) = 0$, $i \in \{1, \ldots, n\}$;
2. $Q = \sum_j Q_j(x) y^j$ avec $\deg Q_i < n - \tau - (k-1)i$.

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Algorithme de Guruswami-Sudan

Constantes :

q, $n < q$, k, et $\tau > t$

$(x_1, \ldots, x_n) \in \mathbb{F}_q$, tous distincts. s un ordre de multiplicité

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[x]$, $\deg f(x) < k$, $d(\mathrm{ev}\, f, y) \leq t$

Interpolation : Trouver $Q \in \mathbb{F}_q[x, y]$

1. $\mathrm{mult}(Q, (x_i, y_i) \geqslant s,\ i \in \{1, \ldots, n\}$;
2. $Q = \sum_j Q_j(x) y^j$ avec $\deg Q_i < s(n - \tau) - (k - 1)i$.

Recherche de racines : Retourner les f tels que $Q(x, f(x)) = 0$.

Retirer les f tels que $f \notin \mathbb{F}_q[x]$ ou $\deg f \geqslant k$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Correction

Définition

$Q \in \mathbb{F}_q[x, y]$ a une multiplicité s en $(0, 0)$, s'il ne présente pas de termes de degré strictement inférieur à s.
Q a une multiplicité s en (a, b), si le polynôme $Q(X + a, Y + b)$ a un multiplicité supérieure à s en $(0, 0)$.

Proposition

Soit $Q \in \mathbb{F}_q[x, y]$ tel que $\text{mult}(Q, (a, b)) \geqslant s$. Soit $f \in \mathbb{F}_q[x]$ tel que $f(a) = b$. Alors $(x - a)^s \mid Q(x, f(x))$.

Proposition

Soit f tel que $d(\text{ev}\, f, y) \leqslant \tau$, et $\deg f < k$. Alors $Q(X, f(X)) = 0$.

Analyse

Plus d'inconnues que d'équations :

$$N_Q > \binom{s+1}{2} n \iff \frac{(s(n-\tau)-1)^2}{2(k-1)} > \binom{s+1}{2} n.$$

Il vient

$$\tau < n - \sqrt{(1+\frac{1}{s})(k-1)n} - \frac{1}{s}.$$

Quand $s = 1$, on retrouve

$$n - \sqrt{2(k-1)n} - 1.$$

Courbes

$$\tau < n - \sqrt{(1 + \frac{1}{s})(k-1)n} - \frac{1}{s}$$

- Quand s est assez grand on obtient $\tau < n - \sqrt{(k-1)n}$. Rayon de Johnson !

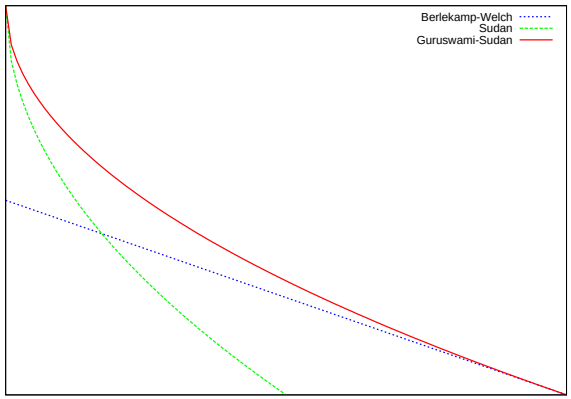


FIG. 6: Rayons $\delta = \frac{\tau}{n}$ en fonction du taux de transmission $R = \frac{k}{n}$

Sudan pour les codes C_L

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,
$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[\mathcal{C}]$, $f \in L(rQ)$, telles que $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q = \sum_{i=0}^{l} Q_i Y^i \neq 0 \in \mathbb{F}_q[C][Y]$ tel que

1. $Q(y_i)(P_i) = 0$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((n - \tau - 1 - ri)Q)$;

Recherche de racines : Retourner les f tels que $Q(f) = 0$.

Retirer les f tels que $f \notin L(rQ)$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Guruswami-Sudan pour les codes C_L

Constantes :

$\mathcal{C}$ une courbe affine en position spéciale, de genre g,
$P_1, \ldots, P_n \in \mathcal{C}(\mathbb{F}_q)$, r un entier définissant le code $C_L(\mathcal{P}, rQ)$.
s un ordre de multiplicité

Entrée :

$y = (y_1, \ldots, y_n)$ le mot reçu.

Sortie : Liste des $f \in \mathbb{F}_q[\mathcal{C}]$, $f \in L(rQ)$, telles que $d(\mathrm{ev}\, f, y) \leq \tau$

Interpolation : Trouver $Q = \sum_{i=0}^{l} Q_i Y^i \neq 0 \in \mathbb{F}_q[C][Y]$ tel que

1. $\mathrm{mult}(Q(y_i, P_i)) \geqslant s$, pour $i \in \{1, \ldots, n\}$.
2. $Q_i \in L((s(n-\tau) - 1 - ri)Q)$;

Recherche de racines : Retourner les f tels que $Q(f) = 0$.

Retirer les f tels que $f \notin L(rQ)$ ou $d(\mathrm{ev}\, f, y) > \tau$.

Si la liste est vide ; échec du décodage

Résultat

Définition de la « multiplicité »

En développant relativement à y_j, on écrit

$$Q(y) = \sum_j (y - y_i)^j Q_{ij}, \quad Q_{ij} \in \mathbb{F}_q(\mathcal{C}),$$

et la multiplicité est définie par

$$\text{mult}(Q, (y_i, P_i)) = \min_j \{ v_{P_i}(Q_{ij}) + j \}.$$

On obtient $\tau \leq n - \sqrt{rn(1 + \frac{1}{s}) - \frac{g}{s}}$, et quand s croît

$$\tau \quad \leq \quad n - \sqrt{rn}$$

Résultat

Définition de la « multiplicité »

En développant relativement à y_j, on écrit

$$Q(y) = \sum_j (y - y_i)^j Q_{ij}, \quad Q_{ij} \in \mathbb{F}_q(\mathcal{C}),$$

et la multiplicité est définie par

$$\text{mult}(Q, (y_i, P_i)) = \min_j \{ v_{P_i}(Q_{ij}) + j \}.$$

On obtient $\tau \leq n - \sqrt{rn(1 + \frac{1}{s})} - \frac{g}{s}$, et quand s croît

$$\begin{aligned} \tau &\leq n - \sqrt{rn} \\ &= n - \sqrt{n(n-d)} \end{aligned}$$

Résultat

Définition de la « multiplicité »

En développant relativement à y_j, on écrit

$$Q(y) = \sum_j (y - y_i)^j Q_{ij}, \quad Q_{ij} \in \mathbb{F}_q(\mathcal{C}),$$

et la multiplicité est définie par

$$\mathrm{mult}(Q, (y_i, P_i)) = \min_j \{ v_{P_i}(Q_{ij}) + j \}.$$

On obtient $\tau \leq n - \sqrt{rn(1 + \frac{1}{s})} - \frac{g}{s}$, et quand s croît

$$\begin{aligned} \tau &\leq n - \sqrt{rn} \\ &= n - \sqrt{n(n-d)} \\ &= \text{Rayon de Johnson !} \end{aligned}$$

Codes CRT

On a l'analogie

- Reed-Solomon (codage des polynômes de degré inférieur à k)

$$\begin{array}{rccc} \text{ev}: & \mathbb{F}_q[X]_k & \to & \mathbb{F}_q^n \\ & f & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

  Distance minimale $d = n - k + 1$.

- Codes CRT (codage des entiers inférieurs à $p_1 \cdots p_k$).

$$\begin{array}{rccc} \text{ev}: & \{f;\ f < K\} & \to & \mathbb{Z}/p_1\mathbb{Z} \times \cdots \times \mathbb{Z}/p_n\mathbb{Z} \\ & f & \mapsto & (f \bmod p_1, \ldots, f \bmod p_n) \end{array}$$

- La distance de Hamming est $d(x, y) = \sum_i \delta(x_i, y_j)$.

Proposition

La distance minimale est $d \geq n - k + 1$.

Décodage en liste

Soit le code CRT donné par :

$$\begin{array}{rccl} \text{ev} : & \{f;\ n < K\} & \to & \mathbb{Z}/p_1\mathbb{Z} \times \cdots \times \mathbb{Z}/p_n\mathbb{Z} \\ & f & \mapsto & (f \bmod p_1, \ldots, f \bmod p_n) \end{array}$$

Soit $(y_1 \bmod p_1, \ldots, y_n \bmod p_n)$ « le mot reçu ».

Théorème

Soit $F \in \mathbb{N}$ auxiliaire. Soit $Q = \sum Q_i y^i \neq 0$ tel que

1. $Q(y_i) = 0 \bmod p_i$.
2. $\log |Q_i| < F - i \log K$

Alors les f cherchés vérifient $Q(f) = 0$.

LLL a la place du système linéaire.
Cet algorithme décode jusqu'à

$$n - \sqrt{2kn \frac{\log p_n}{\log p_1}}$$

Décodage en liste

Soit le code CRT donné par :

$$\begin{array}{rccl} \text{ev} : & \{f;\ n < K\} & \to & \mathbb{Z}/p_1\mathbb{Z} \times \cdots \times \mathbb{Z}/p_n\mathbb{Z} \\ & f & \mapsto & (f \bmod p_1, \ldots, f \bmod p_n) \end{array}$$

Soit $(y_1 \bmod p_1, \ldots, y_n \bmod p_n)$ « le mot reçu ».

Théorème

Soit $F \in \mathbb{N}$ auxiliaire. Soit $Q = \sum Q_i y^i \neq 0$ tel que

1. $Q(y_i) = 0 \bmod p_i{}^s$.
2. $\log |Q_i| < sF - i \log K$

Alors les f cherchés vérifient $Q(f) = 0$.

LLL a la place du système linéaire.
Cet algorithme décode jusqu'à

$$n - \sqrt{kn \frac{\log_n}{\log p_1}}$$

Codes : et la taille du corps ?

Ces algorithmes sont ignorants du corps de base !

- La borne de Johnson

$$J(n,d) \leq n\left(1-\sqrt{1-\frac{d}{n}}\right);$$

est générique relativement à la taille du corps, même infini.

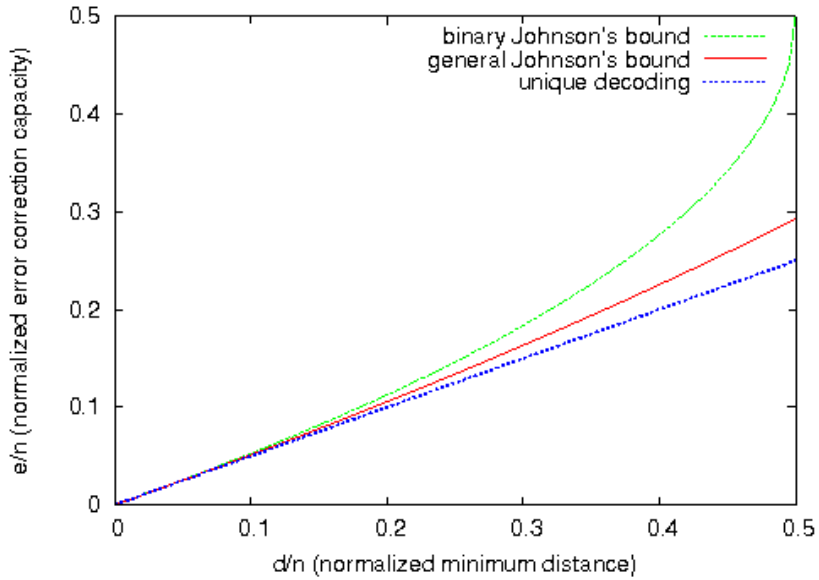
- pour un corps de taille q, si on note $\theta_q = 1 - \frac{1}{q}$, alors

$$J_q(n,d) = \theta_q \cdot n\left(1-\sqrt{1-\frac{1}{\theta_q}\cdot\frac{d}{n}}\right);$$

- dans le cas binaire, on obtient

$$J_2(n,d) = \frac{n}{2}\left(1-\sqrt{1-2\frac{d}{n}}\right).$$

Borne de Johnson binaire versus borne de Johnson générique

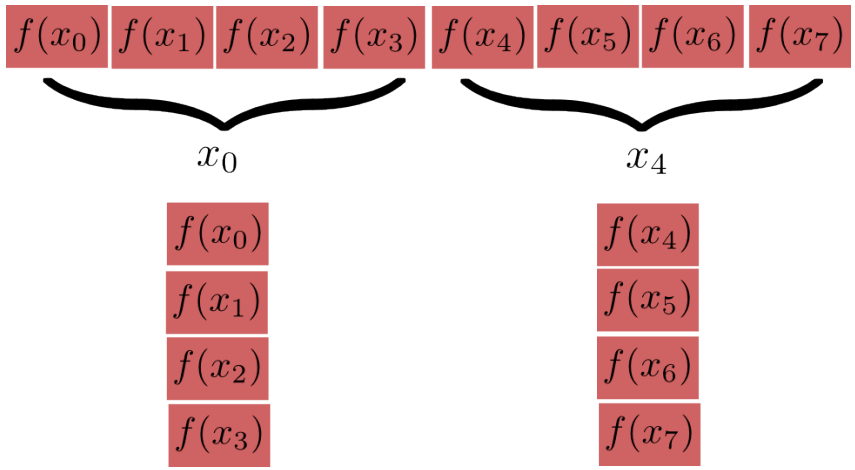


Le rayon de Johnson binaire est obtenu algorithmiquement pour les BCH binaires.

> *Y. Wu. New list decoding algorithms for Reed-Solomon and BCH codes.* IEEE Trans. on Inf. Theory, *2008*

Gros alphabets : codes de Reed-Solomon repliés

On suppose $x_i = \alpha^i$.



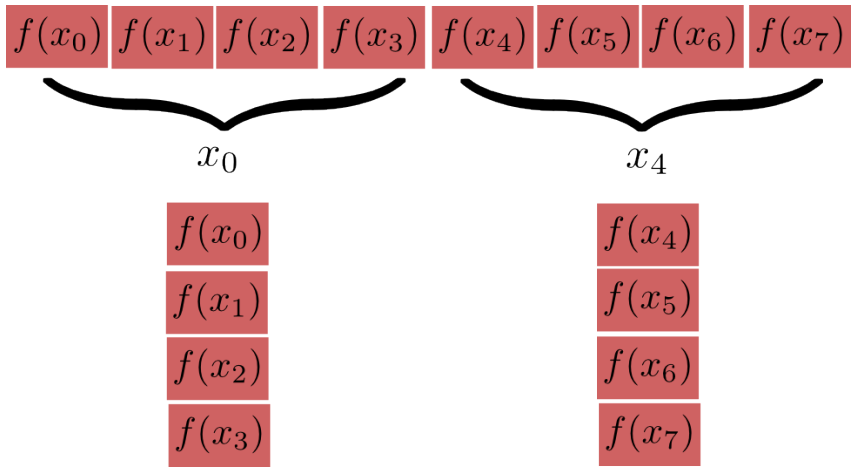
On obtient un code

- sur un alphabet q^4 ;
- non linéaire ;
- de longueur $n/4$;
- de cardinal $q^k = (q^4)^{\frac{k}{4}}$;
- donc de même « taux »

$$\log |C|/N = k/n$$

Décodage

On suppose $x_i = \alpha^i$.



- « Interpolation » avec un polynôme $Q(x, y_1, y_2, y_3, y_4)$;
- « Recherche de racines » : les polynômes f tels que

$$Q(x, f(x), f(\alpha x), f(\alpha^2 x), f(\alpha^3 x)) = 0;$$

- Factorisation possible car

$$f(\alpha x) = f(x)^q \bmod \left(X^{q-1} - \alpha\right);$$

Résultat

Théorème

Pour tout $\varepsilon > 0$, *et* $R \in]0,1[$, *il existe une famille de codes de Reed-Solomon repliés de longueur* $N \longrightarrow \infty$,

- *de taux de transmission* R,
- *qui corrigent en liste au moins* $1 - R - \varepsilon$ *erreurs ;*
- *en temps* $\left(\frac{N}{\varepsilon^2}\right)^{O(\varepsilon^{-1}\log(1/R))}$.

Résultat

Théorème

Pour tout $\varepsilon > 0$, *et* $R \in]0,1[$, *il existe une famille de codes de Reed-Solomon repliés de longueur* $N \longrightarrow \infty$,

- *de taux de transmission* R,
- *qui corrigent en liste au moins* $1 - R - \varepsilon$ *erreurs ;*

- *en temps* $\left(\frac{N}{\varepsilon^2}\right)^{O(\varepsilon^{-1}\log(1/R))}$.
- *la taille de l'alphabet est*

$$\Omega\left(\left(\frac{N}{\varepsilon^2}\right)^{O(\frac{1}{\varepsilon^2})}\right).$$

V. Guruswami and A. Rudra. Explicit capacity-achieving list-decodable codes. In STOC '06

Résultat

Théorème

Pour tout $\varepsilon > 0$, *et* $R \in]0,1[$, *il existe une famille de codes de Reed-Solomon repliés de longueur* $N \longrightarrow \infty$,

- *de taux de transmission* R,
- *qui corrigent en liste au moins* $1 - R - \varepsilon$ *erreurs ;*
  C'est la même capacité que le canal à effacement !
- en temps $\left(\frac{N}{\varepsilon^2}\right)^{O(\varepsilon^{-1}\log(1/R))}$.
- la taille de l'alphabet est

$$\Omega\left(\left(\frac{N}{\varepsilon^2}\right)^{O(\frac{1}{\varepsilon^2})}\right).$$

V. Guruswami and A. Rudra. Explicit capacity-achieving list-decodable codes. In STOC '06

Les codes géométriques à la rescousse

Théorème

Pour tout $\varepsilon > 0$, il existe une famille de code géométriques corrigeant une fraction $1 - \varepsilon$ d'erreurs, en temps polynomial,

- *avec un alphabet constant de taille*

$$Q = \left(\frac{1}{\varepsilon}\right)^{O(\log(\frac{1}{\varepsilon}))}$$

- *et un taux de transmission de*

$$\Omega(\varepsilon)$$

En utilisant la tour de Garcia et Stichtenoth.

Les codes géométriques à la rescousse

Théorème

Pour tout $\varepsilon > 0$, il existe une famille de code géométriques corrigeant une fraction $1 - \varepsilon$ d'erreurs, en temps polynomial,

- *avec un alphabet constant de taille*

$$Q = \left(\frac{1}{\varepsilon}\right)^{O(\log(\frac{1}{\varepsilon}))}$$

- *et un taux de transmission de*

$$\Omega(\frac{\varepsilon}{\log(1/\varepsilon)})$$

En utilisant la tour de Garcia et Stichtenoth.

Hamming versus Shannon

Pour le canal binaire symétrique

$$\Pr(1|0) = \Pr(0|1) = p,$$

on a $C = 1 - H_2(p)$. Donc le théorème de Shannon dit qu'on peut communiquer avec un rendement $R < 1 - H_2(p)$.

Théorème

Pour tout p, pour tout l, et pour n assez grand, il existe un code binaire décodable en liste de taille l, et taux $1 - H_2(p) - 1/l$.

$\Longrightarrow$ On approcherait la capacité même pour le canal adversaire.

Hamming versus Shannon

Pour le canal binaire symétrique

$$\Pr(1|0) = \Pr(0|1) = p,$$

on a $C = 1 - H_2(p)$. Donc le théorème de Shannon dit qu'on peut communiquer avec un rendement $R < 1 - H_2(p)$.

Théorème

Pour tout p, pour tout l, et pour n assez grand, il existe un code binaire décodable en liste de taille l, et taux $1 - H_2(p) - 1/l$.

$\Longrightarrow$ On approcherait la capacité même pour le canal adversaire.

Pas de construction explicite.

Hamming versus Shannon

Pour le canal binaire symétrique

$$\Pr(1|0) = \Pr(0|1) = p,$$

on a $C = 1 - H_2(p)$. Donc le théorème de Shannon dit qu'on peut communiquer avec un rendement $R < 1 - H_2(p)$.

Théorème

Pour tout p, pour tout l, et pour n assez grand, il existe un code binaire décodable en liste de taille l, et taux $1 - H_2(p) - 1/l$.

$\Longrightarrow$ On approcherait la capacité même pour le canal adversaire.

Pas de construction explicite.

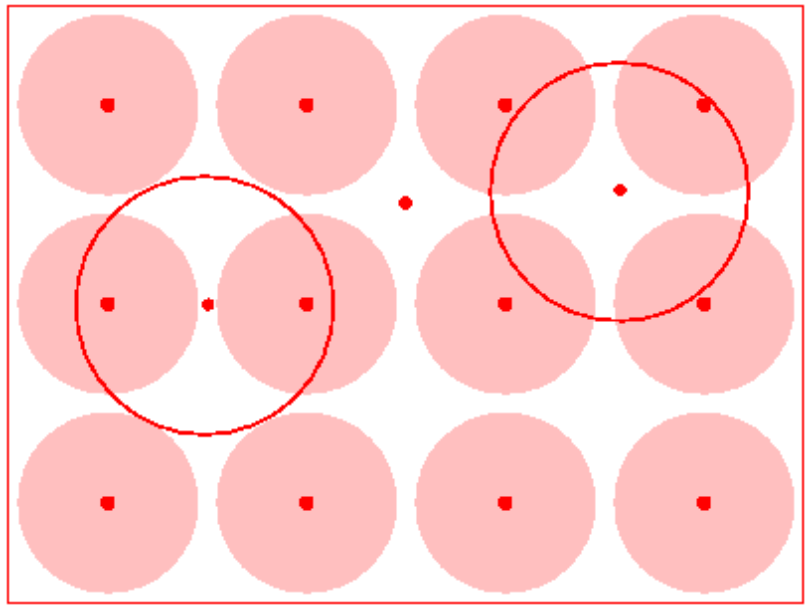
Quelques espoirs du coté des « codes concaténés ».

Idiot, le décodage en liste ?

Mais que faire d'une liste en pratique ?

Idiot, le décodage en liste ?

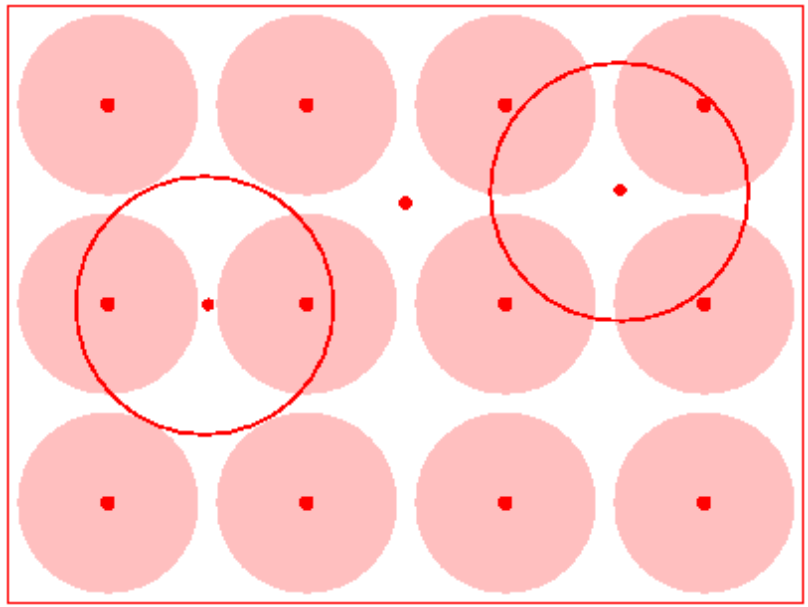
Mais que faire d'une liste en pratique ?



- Selon toute probabilité, la taille de la liste est 1 (démontré pour les codes de Reed-Solomon).

Idiot, le décodage en liste ?

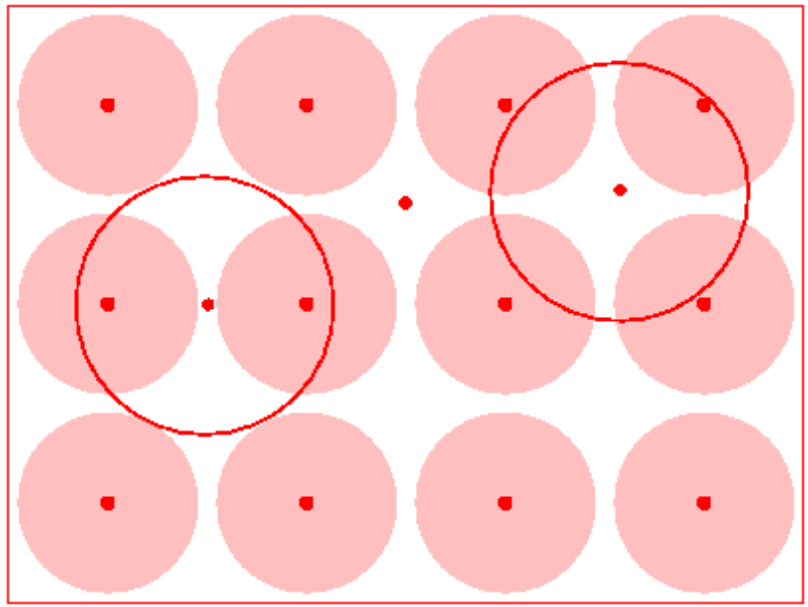
Mais que faire d'une liste en pratique ?



- Selon toute probabilité, la taille de la liste est 1 (démontré pour les codes de Reed-Solomon).
- Les algorithmes sont conçus pour le pire cas. Ils ne sont pas adaptatifs.

Idiot, le décodage en liste ?

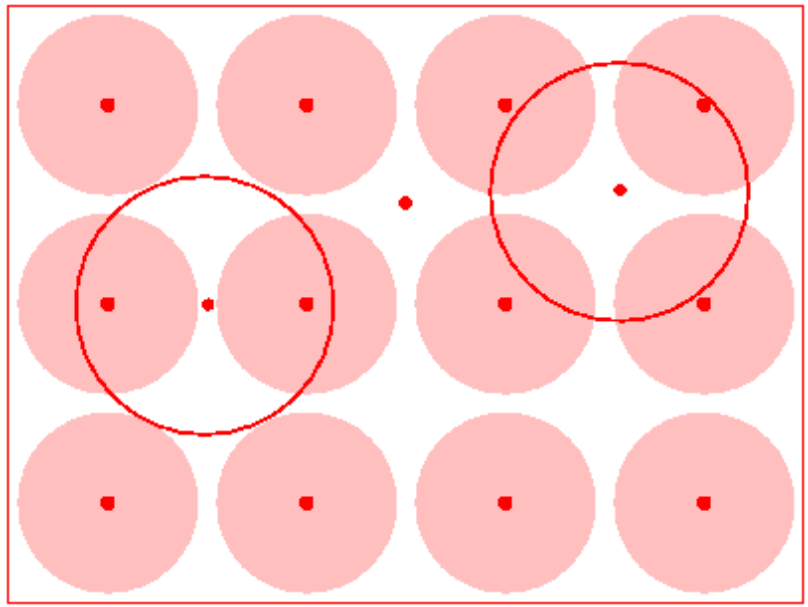
Mais que faire d'une liste en pratique ?



- Selon toute probabilité, la taille de la liste est 1 (démontré pour les codes de Reed-Solomon).
- Les algorithmes sont conçus pour le pire cas. Ils ne sont pas adaptatifs.
- Meilleure complexité $O(l^5 n^2) =$

Idiot, le décodage en liste ?

Mais que faire d'une liste en pratique ?



- Selon toute probabilité, la taille de la liste est 1 (démontré pour les codes de Reed-Solomon).
- Les algorithmes sont conçus pour le pire cas. Ils ne sont pas adaptatifs.
- Meilleure complexité $O(l^5 n^2) = O(ls^4 n^2)$.

Conclusion (I)

Décodage par syndrome.

- « équation clé » ;
- relations de récurrence, 1D ou nD ;
- compliqué pour les codes géométriques ;
- efficace, en arithmétique classique ;
- Berlekamp-Massey bien implanté en matériel ;
- Berlekamp-Massey-Sakata étudié pour le matériel (pipeliné !).

Conclusion (II)

Décodage par interpolation

- décode beaucoup plus d'erreurs ;
- permet le décodage souple (pas de notion d'erreur additive) ;

  R. Kötter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. IEEE Trans. on Inf. Theory, *2003*

- principe géneral, s'applique à de nombreux contextes (codes CRT) ;
- lourd, très lourd (multiplicités) ;
  - Passage en algorithmique rapide $\Longrightarrow$ codes géométriques ? (Difficulté de la tour de Garcia-Stichtenoth, qui n'a pas de modèle plan agréable)
  - Meilleures mises en équations (Wu) ?

Conclusion (II)

Décodage par interpolation

- décode beaucoup plus d'erreurs ;
- permet le décodage souple (pas de notion d'erreur additive) ;

  R. Kötter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. IEEE Trans. on Inf. Theory, *2003*

- principe géneral, s'applique à de nombreux contextes (codes CRT) ;
- lourd, très lourd (multiplicités) ;
  - Passage en algorithmique rapide $\Longrightarrow$ codes géométriques ? (Difficulté de la tour de Garcia-Stichtenoth, qui n'a pas de modèle plan agréable)
  - Meilleures mises en équations (Wu) ?

De beaux progrès théoriques (Guruswami), de nombreux progrès à faire pour être compétitif en pratique.