Analyse Statistique des propriétés métriques des codes linéaires de rendement $R = \frac{1}{2}$.

Travaux-Pratiques I

February 12, 2007

Un $[n, k]$-code linéaire est dit *systématique* quand il posssède une matrice génératrice de la forme $G = (I_k A)$ òu I_r désigne la matrice identité $k \times k$ et A une matrice $k \times (n - r)$ arbitraire. L'objectif de cet exercice de travaux-pratiques est de faire une analyse statistique des propriétés métriques des codes linéaire systématiques de rendement $\frac{1}{2}$, donc des codes de dimension k, et de longueur $2k$. Le plus célèbre des codes de ce type est le Code de Golay étendu $[24, 12, 8]$ dont la matrice génératrice ci-dessous a été obtenue par la commande `GolayCode( GF(2), true)` du programme de calcul formel australien `Magma` disponible en ligne :

1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1
0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	1	1
0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	1

La distribution de poids comprend 1 mot de poids 0, 759 de

1 Implantation

Pour des raisons d'explosions combinatoires, on se limite aux codes de longueur < 64, ainsi, les mots pourront être représentés par des entiers non signés de 64 bits et une matrice génétrice par un pointeur vers un mot.

1. Ecrire une fonction `mot *randcode( int k)` qui renvoie une matrice aléatoire systématique k lignes, $2k$ colonnes.

```
void enum( mot x, mot* G, int k)
{
if ( k ) {
   enum( x, G, k-1 );
   x = x ^ G[ k-1 ];
   enum( x, G, k-1 );
} else {
  traitement du mot x
}
```

```
int main( int argc, char* argv[])
{ int k, cpt;
  mot * G;
  k = atoi( argv[1] );
  if ( argc > 2 ) cpt = atoi( argv[2] );
  else cpt = fct(k);
  while ( cpt-- ) {
     G := RandCode( k );
     ...
     enum( 0, G, k );
     ...
     free( G );
 }
 ...
}
```

Figure 1: Structures des fonctions fondamentales

2. Implanter l'algorithme récursif.
3. Estimer le nombre de codes N_k que votre programme pourra énumérer en moins de 5 minutes.

k	4	8	12	16	20	24	28	32
N_k								

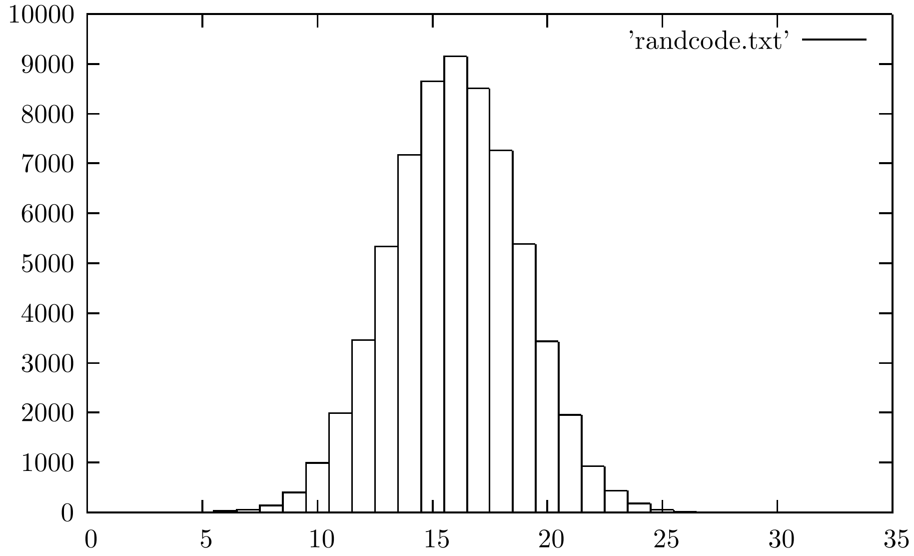
4. Calculer le nombre de total de codes systématiques de rendement $\frac{1}{2}$.
5. Comparer au nombre de codes énumérables en 5 minutes.

2 Distances minimales

1. Modifier votre programme pour calculer la distance minimale.
2. Ecrire un programme qui, pour k donné, calcule la répartition des distances minimales d'un échantillon de codes aléatoires.
3. On note d_k la plus grande distance minimale obtenue sur un échantillon de codes aléatoires. Faire la représentation graphique de la fonction $k \mapsto d_k$.

3 Distribution de poids moyenne

La distribution de poids d'un $[n, k]$-code C est un vecteur de $n + 1$ entiers A_w, $w = 0, 1, \ldots n$ où A_w désigne le nombre de mots de poids w dans C.



1. Modifier votre programme d'énumération pour qu'il retourne la distribution de poids.
2. Ecrire un programme qui calcule la distribution moyenne d'un échantillon de code aléatoire.
3. Faire une représentation graphique.

4 Liens utiles

1. http://magma.maths.usyd.edu.au/magma/htmlhelp/text1558.htm
2. http://magma.maths.usyd.edu.au/calc/
3. La table des meilleurs codes A. Brouwer http://www.win.tue.nl/ãeb/voorlincod.html