ПРЕСМЯТАНЕ НА МИНИМАЛНОТО РАЗСТОЯНИЕ НА ЛИНЕЕН КОД
ON THE CALCULATION OF THE MINIMUM DISTANCE OF A LINEAR CODE

Iliya Bouyukliev
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
iliyab@math.bas.bg

Abstract
Some aspects of the algorithms for calculating the minimum distance of linear codes over finite fields are presented.

Keywords: Linear codes, Minimum distance, Algorithms.

ВЪВЕДЕНИЕ

Минималното разстояние на линеен код е една от най-важните му характеристики. Но намирането му по дадена пораждаща матрица е NP пълна задача [2].

Известните досега практически алгоритми се базират на максимален брой пораждащи матрици $G_1, G_2, G_3, \ldots, G_s$ с непресичащи се множества от систематични координати. Такива са алгоритмите на Брауер и Цимерман и различни техни модификации за циклични кодове, квази-циклични кодове, кодове с кратни на цяло число тегла и др. Те са описани в [1], [3], [7], [8] и имплементирани в софтуерните пакети MAGMA [4] и GAP [5]. Основната идея е, че след като се направят 1,2,…, l линейни комбинации на редовете на всички матрици, всички кодови думи с тегло w (зависещо от l) ще бъдат генерирани при условие, че има такива. И ако до този момент на търсене най-леката генерирана кодова дума е с тегло $w+1$, то минималното разстояние на кода $d(C)$ е равно на $w+1$.

Въпросът, който разглеждаме тук, е кои матрици измежду $G_1, G_2, G_3, \ldots, G_s$ да използваме и кои кодови думи за всяка от матриците да генерираме, така че да обходим най-малко вектори.

ИЗЛОЖЕНИЕ

Нека F_q^n е n-мерно векторно пространство над крайно поле F_q. Всяко k – мерно подпространство C на F_q^n наричаме линеен $[n,k]$ код с дължина n и размерност k, а векторите от C наричаме кодови думи. Тегло (по Хеминг) за кодова дума $wt(c)$ дефинираме като брой ненулеви координати. За линеен код, минималното измежду теглата на кодовите думи съвпада с минималното разстояние.

Нека C е линеен $[n,k]$ код с дължина n и размерност k. Множеството $T \subset \{1,2,\ldots,n\}$ с мощност $|T| \leq k$ наричаме систематично множество, ако съответните стълбове в пораждаща матрица на кода са линейно независими. Ако $|T| = k$ множеството се нарича *пълно,* а в противен случай *частично*. Тогава съществува и систематична матрица G

(зависеща от T), такава че стълбовете в G, определени от елементите на T, образуват единична матрица $I_{|T|}$.

Да разгледаме случая, когато n е кратно на k, $t = \frac{n}{k}$ и C има систематични множества $T_1, T_2, T_3, \ldots, T_t$, които не се пресичат. Такива са например самодуалните кодове или t-CIS (complementary information set) кодовете. Тогава е ясно, че минималното разстояние е поне t. Това е така, защото за всяка кодова дума, съответстваща на нетривиална линейна комбинация на редовете на коя да е пораждаща матрица, ще има поне по една ненулева координата за всяко систематично множество. Ако намерим теглата на всички редове на G_1 , ще определим и всички кодови думи с тегло t, но не и с тегло $t+1$. Това е така, защото може да съществува кодова дума с две ненулеви координати от T_1 и тя няма да е генерирана. Ако намери теглата и на всички редове на G_2, ще определим и всички кодови думи с тегло t+1 (ако има такива) и т.н.

За да разгледаме общия случай, се нуждаем от допълнителни означения. С $U_j^{(a_i)}$ бележим множеството от всички кодови думи с $j \leq a_i$ на брой ненулеви координати в систематичното множество T_i. За доказване на следващите твърдения ще използваме следния факт, базиран на принципа на Дирихле.

Лема 1.

Ако $m = b_1 + b_2 + \ldots + b_e + e - 1$, то за всяко разбиване на m на сума от e положителни цели числа $d_1 + d_2 + \ldots + d_e$ ще има поне едно $i \leq e$, такова че $d_i \leq b_i$.

При тези означения е в сила следното твърдение:

Лема 2.

Нека $a_1, a_2, a_3, \ldots, a_t$ са цели неотрицателни числа, не по-големи от k, а C е [n,k] код с непресичащи се систематичните множества $T_1, T_2, T_3, \ldots, T_t$, такива че $|T_1| + |T_2| + \ldots + |T_t| = n$ със систематични матрици $G_1, G_2, G_3, \ldots, G_t$ с размерност k. Тогава $U = U_1^{(a_1)} \cup U_2^{(a_2)} \cup \ldots \cup U_t^{(a_t)}$ съдържа всички кодови думи с тегла, не по-големи от $m = a_1 + a_2 + \ldots + a_t + t - 1$.

Доказателство

Нека c е кодова дума с тегло, не по-голямо от m. Тя има $b_1, b_2, b_3, \ldots, b_t$ ненулеви координати в систематичните множества $T_1, T_2, T_3, \ldots, T_t$ такива че $m = b_1 + b_2 + b_3 + \ldots + b_t$. Тогава съществува $i \leq t$ такова че $b_i \leq a_i$ и $c \in U_i^{(a_i)}$. Но c е произволно избрано, което и доказва твърдението.

Тези твърдения определят естествен алгоритъм, който е в основата на целия подход, като се вземе предвид, че в този случай $U_i^{(a_i)}$ съвпада с линейните комбинации на до a_i реда на матрицата G_i. Описание на алгоритъма може да се намери в [1] и [7].

Да разгледаме по-общия случай, когато C има систематични непресичащи се множества $T_1, T_2, T_3, \ldots, T_t, \ldots, T_s$, като първите t са пълни т.е. са с мощност k, а следващите са с мощност по-малка от k. Можем да считаме, че са подредени според мощността им. Без ограничение на общността (с точност до пермутация на координатите) съответните им систематични матрици $G_1, G_2, G_3, \ldots, G_s$ имат вида:

$(I_k|A'_1), (A_2|I_k \quad A'_2), \dots, (A_t|I_k \quad A'_t), \left(A_{t+1}\middle|\begin{matrix} I_{|T_{t+1}|} & A'_{|T_{t+1}|} \\ 0 & 0 \end{matrix}\right), \dots, \left(A_s\middle|\begin{matrix} I_{|T_s|} \\ 0 \end{matrix}\right).$

При $i \le t$ имаме:

$|U_i^{(a_i)}| = (q-1)\binom{|T_i|}{1} + (q-1)^2\binom{|T_i|}{2} + \dots + (q-1)^{a_i}\binom{|T_i|}{a_i}$ и $|U_i^{(0)}| = 0;$

А при $i > t$

$|U_i^{(a_i)}|=$

$\left((q-1)\binom{|T_i|}{1} + (q-1)^2\binom{|T_i|}{2} + \dots + (q-1)^{a_i}\binom{|T_i|}{a_i}\right)\left(q^{(k-|T_i|)}\right) + q^{(k-|T_i|)}.$

В този случай за мощността на имаме $|U_i^{(0)}|=q^{(k-|T_i|)}$.

Лема 3.

Нека C е [n,k] линеен код със систематични непресичащи се множества $T_1, T_2, T_3, \dots, T_t, \dots, T_s$, като първите t са пълни т.е. са с мощност k, а следващите са с мощност, по-малка от k, със систематични матрици $G_1, G_2, G_3, \dots, G_s$. Нека $a_1, \dots, a_t$, $a_{t+1}, \dots, a_{t+r}$, $t + r = s$, са цели неотрицателни числа, по-малки от k, и най-много едно от числата $a_{t+1}, \dots, a_{t+r}$ е нула. Тогава $U = U_1^{(a_1)} \cup U_2^{(a_2)} \cup \dots \cup U_{t+r}^{(a_{t+r})}$ съдържа всички кодови думи с тегла, не по-големи от $m = a_1 + a_2 + \dots + a_t + a_{t+1} + \dots + a_{t+r} + t + r - 1$.

Доказателство

Нека c да е кодова дума с тегло, не по-голямо от m. Тя има $b_1, b_2, b_3, \dots, b_{t+r}$ ненулеви координати в систематичните множества $T_1, T_2, T_3, \dots, T_{t+r}$ такива, че $m = b_1 + b_2 + b_3 + \dots + b_{t+r}$. Нека за $t < j \le t + r$ е изпълнено $b_j = a_j = 0$. В този случай $U_j^{(b_j)}$ съдържа c. Ако c е такова, че няма $b_j = a_j = 0$ за $t < j \le t + r$, този случай се свежда към Лема 2. Това са всички възможности за c, което и доказва твърдението.

За положително δ съществуват цели неотрицателни числа $a_1, \dots, a_t$, $a_{t+1}, \dots, a_{t+r}$, $t + r = s$, по-малки от k+1, като най-много едно от числата $a_{t+1}, \dots, a_{t+r}$ е нула, такива че $\delta = a_1 + a_2 + \dots + a_t + a_{t+1} + \dots + a_{t+r} + t + r - 1$ и

$S = \min_{a_1, a_2, \dots, a_{t+r}; r=1,2,\dots,s-t} \left| U_1^{(a_1)} | + | U_2^{(a_2)}| + \dots | U_t^{(a_t)}| + |U_{t+1}^{(a_{t+1})}| + \dots + |U_{t+r}^{(a_{t+r})} \right|$.

За тези δ и $a_1, \dots, a_t$, $a_{t+1}, \dots, a_{t+r}$ дефинираме

$\Omega_\delta = U_1^{(a_1)} \cup U_2^{(a_2)} \cup \dots \cup U_t^{(a_t)} \cup U_{t+1}^{(a_{t+1})} \cup \dots \cup U_{t+r}^{(a_{t+r})}$.

При тези означения δ е такова, че всички вектори с такова тегло от кода C са в Ω_δ, а ако няма такива вектори, то δ е долна граница $lb = \delta$ за минималното разстояние на C. Най-леката дума в Ω_δ е горна граница ub за минималното разстояние на C. Точната стойност на $d(C)$ се получава, когато $lb+1 = ub$, или когато в процеса на генериране на Ω_δ се намери кодова дума с тегло $lb = \delta$.

Алгоритъм

INPUT: Линеен код C със систематични непресичащи се множества $T_1, T_2, T_3, \ldots, T_t, \ldots, T_s$, като първите t са пълни, т.е. са с мощност k, а следващите са с мощност, по-малка от k (подредени според мощността) и съответните им систематични матрици.

OUTPUT: Минималното разстояние *d(C).*

1. $\delta = t - r$;
2. *ub=n;*
3. *while* ($\delta + 1 <$*ub) do{*
4. $\delta = \delta + 1$;
5. *Generate* Ω_δ *and new ub;}* // ub is the weight of the lightest v in Ω_δ;
6. *Print d(C)=ub.*

Коректността на алгоритъма следва от Лема 3 и въведените означения. За отбелязване е, че Ω_δ, дефинирано от $a_1, \ldots, a_t, a_{t+1}, \ldots, a_{t+r}$ и $\Omega_{\delta+1}$ дефинирано от $b_1, \ldots, b_t, b_{t+1}, \ldots, b_{t+r}$, в общия случай се различават само в една позиция, като някое b_j е с едно повече от съответното a_j или r' е с едно повече от r .

Въпреки че $|\Omega_\delta|$ зависи от много променливи, за определяне на подходящите $a_1, \ldots, a_t, a_{t+1}, \ldots, a_{t+r}$ за намиране на минимума на $|\Omega_\delta|$ е необходимо сравняване само между $\left|U_{t+r+1}^{(0)}\right|$, $\left|U_{t+r+1}^{(1)}\right|$и $\left|U_1^{(a_1)}\right|$.

За по-голяма ефективност генерираме само непропорционални вектори, като всеки нов вектор се получава само със събиране с предходния (виж [2]). Ефективен алгоритъм за намиране на максимални непресичащи се множества е представен в [8].

Пример

Параметри:

[20,8,≤ 7] $|T_1| = 8; |T_2| = 8; |T_3| = 4;$

Стъпки:

$\left|U_1^{(1)}\right| < \left|U_3^{(0)}\right|$.

1. $a_1 = 1;$ $\delta = 2$ $|\Omega_2|$=8
2. $a_1 = 1, a_2 = 1,$; $\delta = 3$ $|\Omega_3|$=16

$\left|U_1^{(2)}\right| > \left|U_3^{(0)}\right|$.

3. $a_1 = 1, a_2 = 1, a_3 = 0$ $\delta = 4$ $|\Omega_4|$=16+16

$\left|U_1^{(2)}\right| < \left|U_3^{(1)}\right|$.

4. $a_1 = 2, a_2 = 1, a_3 = 0$ $\delta = 5;$ $|\Omega_5|$=36+16+16
5. $a_1 = 2, a_2 = 2, a_3 = 0;$ $\delta = 6;$ $|\Omega_6|$=2.36+16+16

В Таблица 1 са представени експериментални резултати, като са сравнени с онлайн калкулатора на MAGMA и пакета QextNewEdition [6]. В първата колона са представени параметрите на кодовете, във втората и четвъртата броят на генерираните вектори за

намиране на минималното разстояние, а в третата и петата - времето за пресмятане в секунди.

Таблица 1. Експериментални резултати

	MAGMA		QextNewEdition	
Параметри	Брой вектори	Сек.	Брой вектори	Сек.
[45,15,16]	1 053 993	0.030	1 173 563	0.026
[44,15,16]	1 822 761	0.037	1 454 639	0.020
[43,15,16]	2 591 529	0.047	2 467 919	0.026
[42,15,15]	2 591 529	0.054	1 581 195	0.015
[41,15,14]	2 591 529	0.040	2 153 977	0.026
[40,15,14]	7 716 649	0.124	1 701 571	0.017
[39,15,12]	1 727 701	0.030	362 729	0.010

ACKNOWLEDGEMENTS

The research of Iliya Bouyukliev was supported, in part, by a Bulgarian NSF contract KP-06-Russia/33/17.12.2020

ЛИТЕРАТУРА

[1] A. Betten, H. Fripertinger, A. Kerber, A. Wassermann and K.-H. Zimmermann, 1998, Codierungstheorie – Konstruktion und Anwendung linearer Codes. Springer-Verlag, Berlin–Heidelberg–New York.

[2] I. Bouyukliev and V. Bakoev, 2005, Algorithms for computing the number of codewords of fixed weight in linear codes, Proc. of IV Intern. Workshop on Optimal codes and related topics, June 17-23, Pamporovo, Bulgaria, 36-41.

[3] A.Canteaut and F. Chabaud, 1998. A New Algorithm for Finding Minimum-Weight Words in a Linear Code: Application to McEliece's Cryptosystem and to Narrow-Sense BCH Codes of Length 511. IEEE Transactions on Information Theory 44(1): 367-378.

[4] Computational Algebra Group at the University of Sydney, The Magma Computational Algebra System, http://magma.maths.usyd.edu.au/magma/

[5] J. Cramwinckel, E. Roijackers, R. Baart, E. Minkes, L. Ruscio and D. Joyner, GAP package GUAVA, http://cadigweb.ew.usna.edu/ wdj/gap/GUAVA/

[6] I. Bouyukliev, QextNewEdition - GENERATION module, Online available at http://www.moi.math.bas.bg/moiuser/_data/Software/QextNewEdition.html

[7] Grassl M. 2006. Searching for linear codes with large minimum distance. In: Bosma W., Cannon J. (eds) Discovering Mathematics with Magma. Algorithms and Computation in Mathematics, vol 19. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-540-37634-7_13

[8] Lisoněk P, Trummer L. 2016. Algorithms for the Minimum Weight Of Linear Codes. Advances in Mathematics of Communications. Vol. 1; 10(1).