

ABOUT ALGORITHMS FOR CALCULATION OF COVERING RADIUS OF LINEAR CODES

Maria Pashinska-Gadzheva

*Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences, Bulgaria*
mariqpashinska@math.bas.bg

Iliya Bouyukliev

*Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences, Bulgaria*
iliyab@math.bas.bg

ОТНОСНО АЛГОРИТМИ ЗА ПРЕСМЯТАНЕ НА РАДИУС НА ПОКРИТИЕ НА ЛИНЕЙНИ КОДОВЕ

Abstract

The covering radius is an important parameter of linear codes. Its calculation is an NP-complete problem. There are three main approaches to its computing that can be considered. The first method is based on traversing the cosets of the linear code. The second method uses the parity-check matrix of the code to calculate the covering radius. The third method is based on fast Walsh-Hadamard transforms for the binary case and Vilenkin-Chrestenson transform for the non-binary case. In this work, the first two approaches are considered and some of the features of their implementations are presented.

Keywords: Covering Radius; Linear Codes.

ВЪВЕДЕНИЕ

Радиусът на покритие на линеен код е един от параметрите на линейните кодове, които се използват за откриване и поправяне на грешки и компресиране. Той дава информация за възможностите на кода за поправяне на грешки при използването на метода на максималното правдоподобие за декодиране. Тогава радиусът на покритие показва максималното тегло на коригируем вектор-грешка. Намирането му е изчислително тежка задача като съществуват много случаи, в които радиусът на покритие на класове от кодове, не е известен. В тази работа ще бъдат разгледани два основни подхода за намиране на радиус на покритие, които са особено подходящи за паралелно имплементиране чрез векторизация. За описанието на тези алгоритми първо ще представим някои основни дефиниции и теореми.

- Линеен код C се нарича всяко k -мерно подпространство на n -мерното векторно пространство над поле с q елемента F_q^n . Кодовите с тези параметри означаваме като $[n,k]$ кодове.
- Матрица G с k реда и n стълба се нарича пораждаща матрица на линейния код C , ако редовете на G образуват базис на C . Матрицата H с $n-k$ реда и n стълба се нарича проверочна матрица на C , ако за всяка кодова дума $x \in C$ е в сила $Hx^T = 0$.
- Теглото (по Хеминг) $wt(x)$ на вектор $x \in F_q^n$ се нарича броят на неговите ненулеви координати.
- Съседен клас на код C относно вектор $x \in F_q^n$ се нарича множеството $x + C = \{x + c \mid \forall c \in C\}$. Векторното пространство се разделя на $q^{n-k} - 1$ непресичащи се съседни класове, като всеки от тях съдържа q^k вектора. Лидер на съседния клас $x + C$ се нарича вектора с най-малко тегло. Нека s

α_i означим броя на лидерите с тегло i , където $i = 1, 2, \dots, n$. Редицата $(\alpha_1, \dots, \alpha_n)$ се нарича тегловно разпределение на лидерите на съседните класове.

- Радиусът на покритие R на линеен код C се нарича теглото на най-тежкия лидер на съседен клас. Радиусът на покритие на код C може също така да се дефинира като най-малкото естествено число R , такова че сферите с радиус R , описани около кодовите думи на C , покриват цялото пространство F_q^n .

Известни са няколко подхода за намиране на радиуса на покритие на линеен код. Един подход е базиран на тази дефиниция за радиус на покритие, като за целта трябва да бъдат намерени всички лидери на съседни класове и техните тегла. Друг подход се базира на Лема 1.1 от [1], където радиусът на покритие се определя като най-малкото число R , такова че всички $(n-k)$ -мерни вектори могат да бъдат представени като линейна комбинация на не повече от R колони за избрана проверочна матрица на кода. Известен е и друг метод за намирането на радиуса на покритие на един линеен код, който се базира на използването на бързи трансформации [2]. В тази работа ще бъдат разгледани първите два подхода. Тези подходи също така в своята софтуерна имплементация включват извършване на операции с вектори над крайни полета, което ги прави подходящи за използване на векторизация.

ИЗЛОЖЕНИЕ

Нека разгледаме подхода за изчисление на радиуса на покритие, като се използват лидерите на съседните класове. За намирането на радиуса на покритие е необходимо да бъдат обходени всички лидери на съседни класове на кода C , където теглото на най-тежкия от тези лидери е равно на радиуса на покритие. Тогава задачата се свежда до задачата за намиране на тегловното разпределение на съседните класове на линейния код. Естественят подход е да се използва изчерпващо търсене на лидерите на съседните класове. Известно е, че лидерите на съседни класове с тегла до $t = \lfloor \frac{d-1}{2} \rfloor$ са единствени. Също така са известни горни граници на радиусите на покритие [3]. Ако с R_{max} се означа горна граница за радиус на покритие на даден линеен код, търсенето се свежда до намирането на лидерите на съседните класове с лидери с тегла в интервала от t до R_{max} . За целта може да разгледаме следното твърдение:

Тв. 1: Нека C е $[n, k]$ код над поле с q елемента с пораждаща матрица $G = (I_k | A)$. Тогава всеки съседен клас на C съдържа единствен вектор от вида $v = (v_1, \dots, v_n)$, където $v_i = 0$ за $i=1, \dots, k$.

Доказателство: Нека разгледаме $v \in x + C$, където $v_i = \alpha, \alpha \in F_q$ и $g = \alpha g_i$, където g_i е реда i на матрицата G . Тогава вектора $v' = v - g$ е от същия съседен клас, където $v'_i = 0$. Нека предположим, че векторите $u' = y + c'$ и $u'' = y + c''$ са в един и същ съседен клас $y + C$ и първите k координати са нулеви. Тогава вектора $c = u' - u'' = y + c' - y - c'' = c' - c''$ е кодова дума с нули в първите k координати. Това е невъзможно когато $c \neq 0$.

Благодарение на Твърдение 1 можем да намерим представител на всеки съседен клас, като за всеки съседен клас е необходимо да бъдат намерен негов лидер. Броят на тези ненулеви вектори е $q^{(n-k)} - 1$. Следващото твърдение ни позволява да сведем обхождането само до непропорционални лидери.

Тв. 2: Нека l е лидер на съседен клас на кода C . Тогава векторът al ще бъде лидер на съседния клас $(al + C)$.

Доказателство: Нека допуснем, че съществува вектор $al + y \in (al + C), y \neq 0$, който е лидер на съседния клас. Следователно $wt(al + y) < wt(al)$. Ако умножим този

вектор с елемента $\alpha^{-1} \in F_q$, ще получим $\alpha^{-1}al + \alpha^{-1}y = l + \alpha^{-1}y \in (l + C)$, тъй като $\alpha^{-1}y \in C$. В сила е $wt(al + y) = wt(\alpha^{-1}al + \alpha^{-1}y)$, но от това, че l е лидер на съседния клас $(l + C)$ следва $wt(l) = wt(al) < wt(al + y)$. Следователно има противоречие с допускането и al е лидера на съседния клас $(al + C)$.

Тъй като при зададена пораждаща матрица $G = (I_k|A)$, всеки вектор от вида $v = (v_1, \dots, v_n)$, където $v_i = 0$ за $i=1, \dots, k$ задава съседен клас (Тв. 1) и следователно лидер на съседен клас, за намирането на радиуса на покритие на линейен код с пораждаща матрица G е необходимо да бъдат обходени само $\theta = \frac{q^{n-k}-1}{q-1}$ непропорционални вектори.

Лидерът на съседния клас може да бъде представен като сбор от линейна комбинация на редовете на G и вектора v . Оптимизирането на този подход се свежда до намаляване на броя на линейните комбинации, които трябва да бъдат изчислени. Такъв подход е представен в [4], като тук ще представим основната идея за линейни кодове, където $k = n/2$.

Нека разгледаме вектора $v = (v_1, \dots, v_n) \in x + C$, където $v_i = 0$ за $i=1, \dots, k$ и пораждащи матрици $G_1 = (I_k|A_1)$ и $G_2 = (A_2|I_k)$ на кода C . Тогава v и G_2 определят вектора $v' = (v'_1, \dots, v'_n)$ от същия съседен клас, където $v'_i = 0$ за $i=k+1, \dots, n$. Нека x е лидер на съседния клас. Тогава $wt(x) \leq R_{max}$ и x няма да има повече от $\lfloor \frac{\theta R_{max}}{2} \rfloor$ ненулеви стойности в първите или вторите k координати. Всички такива вектори могат да бъдат намерени като сбор на вектора v с линейна комбинация на не повече от $\lfloor \frac{\theta R_{max}}{2} \rfloor$ реда на G_1 или сбор на вектора v' с линейна комбинация на не повече от $\lfloor \frac{\theta R_{max}}{2} \rfloor$ реда на G_2 . Ако се намери вектор с тегло $s < R_{max}$, търсенето продължава за $R_{max} = s$. Също така векторите с тегло по-малко от t не е необходимо да бъдат обхождани, тъй като те определят единствени лидери на съседни класове. Този подход може да се разшири за произволно $k \neq n/2$, като се използват матрици от вида $(I_k|A_1| \dots | A_s)$, $(A_1'|I_k| \dots | A_s')$, където A_i и A_i' , $i = 1, \dots, s$ са $k \times k$ квадратни матрици.

Вторият подход за намиране на радиус на покритие използва проверочната матрица на кода. На всеки съседен клас, определен от вектор x и пораждаща матрица G , можем да съпоставим единствен $(n-k)$ -мерен вектор $s = Hx^T$, наречен *синдром на съседния клас*. Тогава синдромът на един съседен клас може да се представи като линейна комбинация на стълбовете на проверочната матрица. Нека един $[n, k]$ код C има радиус на покритие R . Съседните класове, определени от вектори с тегло по-малко или равно на R , трябва да покрият цялото векторно пространство F_q^n . Тъй като на всеки съседен клас съпоставяме $(n-k)$ -мерен синдром, линейните комбинации на $i = 1, \dots, R$ стълба на пораждащата матрица ще покрият всички $(n-k)$ -мерни вектори. С други думи, за да се намери радиуса на покритие линейен $[n, k]$ код с пораждаща матрица G и проверочна матрица H , относно G , трябва да бъдат намерени линейни комбинации на i стълба на H , като i се увеличава докато всички синдроми не бъдат обходени. Знаем, че броят на синдромите и броят на съседните класове е $q^{n-k} - 1$. Нека означим с $h_1, \dots, h_n$ стълбовете на проверочната матрица и разгледаме синдрома $s_1 = Hx^T = x_1h_1 + \dots + x_nh_n$. Вектора αx определя друг съседен клас със синдром $s_2 = H\alpha x^T = \alpha x_1h_1 + \dots + \alpha x_nh_n = \alpha(x_1h_1 + \dots + x_nh_n) = \alpha s_1$. Тогава за намирането на радиуса на покритие е необходимо да бъдат обходени само непропорционалните синдроми, които са $\frac{q^{n-k}-1}{q-1}$.

ЗАКЛЮЧЕНИЕ

Двата представени подхода за намиране на радиус на покритие на линеен код са от особен интерес за последваща софтуерна имплементация, като се използва векторизация и разширените векторни регистри на съвременните централни процесори. В [5] са представени имплементации на векторизирани алгоритми за събиране на вектори над крайни полета с не повече от 128 елемента като се наблюдават до 12 пъти по-бързи изчисления в сравнение със софтуерния пакет Magma и над 2000 пъти по-бързи изчисления в сравнение с GAP като се използват разширени регистри с дължина 512 бита и AVX512 инструкции за x86 архитектури. Тъй като изчислителната част в двата алгоритъма включва намирането на линейни комбинации на редове или стълбове на матрици, близки ускорения на векторизирани имплементации на тези алгоритми биха били очаквани.

БЛАГОДАРНОСТИ

Работата на авторите е частично подкрепена от Националната пътна карта за РИ, финансирана от МОН на Република България (договор № Д01-325/01.12.2023г.).

REFERENCES // ЛИТЕРАТУРА

1. T. Helleseth, “On the covering radius of cyclic linear codes and arithmetic codes”, *Discrete Applied Mathematics*, vol. 11 (1985), pp 157 - 173.
2. P. Piperkov, I. Bouyukliev, St. Bouyuklieva, “An Algorithm for Computing the Covering Radius of a Linear Code Based on Vilenkin-Chrestenson Transform”, *New Horizons in Differential Geometry and its Related Fields*, 2022, pp. 105 - 123
3. G. Cohen, I. Honkala, S. Litsyn and A. Lobstein, “Covering Codes”, North-Holland, Elsevier Science B.V., 1997.
4. I. Bouyukliev, I. Gancheva, “A method for calculation the weight distribution of the coset leaders of a linear code”, *Algebraic and Combinatorial Coding Theory 2004*, Kranevo, Bulgaria, 2004, pp. 67-72.
5. M. Pashinska-Gadzheva, I. Bouyukliev, “About Methods of Vector Addition over Finite Fields Using Extended Vector Registers”. In: Lirkov, I., Margenov, S. (eds) *Large-Scale Scientific Computations. LSSC 2023. Lecture Notes in Computer Science*, vol 13952. Springer, Cham, 2024, pp. 427-434.

Received: 15-08-2024 Accepted: 24-09-2024 Published: 20-12-2024

Cite as:

Pashinska-Gadzheva, M., Bouyukliev, I. (2024). “About Algorithms for Calculation of Covering Radius of Linear Codes”, *Science Series “Innovative STEM Education”*, volume 06, ISSN: 2683-1333, pp. 95-98, 2024.
DOI: <https://doi.org/10.55630/STEM.2024.0609>