G-VALUES AND DECODING OF GREEDY CODES

Shoaib u Din Department of
Mathematics University of the
Punjab Lahore
shoaibdin@gmail.com

Khalil Ahmad
Department of Mathematics
University Of Management and Technology Lahore
khalil@umt.edu.pk

ABSTRACT

Greedy algorithm is used to develop a class of binary error-correcting codes for given length n and minimum distance d by arranging all strings of length n in B-odering. On the basis of minimum distance d a non negative integer is assigned to each string in the B-odering called g-value [2]. In this paper, We proposed a new algorithm for the allocation of g-values to the binary vectors along with a g-value decoding algorithm for binary linear codes.

Keywords: B-ordering, greedy algorithm, g-value, binary linear codes.

1 INTRODUCTION

C(n,k,d) a binary linear error correcting code is a set of code words, each of length n, contains 2^k code words with the condition that minimum distance between any two code words is greater than or equal to d. There are a number of ways to generate an error correcting code, however in this paper only greedy codes are discussed. Greedy codes are the codes generated by the application of greedy algorithm on the vectors arranged in B-ordering[4]. Each vector is assigned a unique non negative integer known as g-value. All possible vectors of a given length are divided onto various classes according to there g-values. On the basis of properties of g-values an algorithm for decoding is designed.

2 B-ORDERING

Greedy algorithm at any stage makes the choice that appears to be the best at that stage. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution. The choice made by greedy algorithm may depend on previous choices, but it doesn't depend on any future choice.

The application of greedy algorithm requires some sort of ordering in the choices, e.g. in activity selection problem greedy algorithm works with the assumption that input activities are ordered by increasing finishing time. In knapsack problem items are ordered in increasing weight. In greedy codes we apply greedy algorithm on the choices arranged according to B-ordering.

A greedy code is a code generated by the application of a greedy algorithm, where the vectors are arranged in a certain type of ordering. When first vector in the list having a given property is selected, the algorithm continues recursively, until the list is exhausted. The problem is to maximize the number of vectors in the code. The use of greedy algorithm for generating codes is especially attractive due to its natural way of producing a code with a given minimum distance. The vectors are arranged in an ordering and each vector is analyzed in turn for being accepted or rejected according to its distance from vectors already chosen.

The ordering is important and it happens that the B-ordering, a very natural type of ordering, gives a linear code every time.

B-ordering is a generalization of lexicographic ordering[7]. It was first defined and discussed by

Brualdi and Pless in [2]. We get a B-ordering of all binary n-tuples by choosing an ordered basis $\{b_1, b_2, \ldots\ldots b_n\}$ of v_n. The first vector in the B-ordering is the zero vector, and the next is b_1, b_2, $b_2 \oplus b_1, b_3, b_3 \oplus b_1$, $b_3 \oplus b_2$, $b_3 \oplus b_2 \oplus b_1$, b_4, ……where if the first 2^{i-1} vectors of the ordering have been generated using B- basis elements $b_1, b_2, \ldots\ldots b_{i-1}$, then the next 2^{i-1} vectors are generated by adding b_i to those vectors already produced in order.

3 GREEDY CODES

Given a minimum distance d, choose a set of vectors ***C*** with the zero vector first; then go through the vectors in B-ordering and choose the next which has distance d or more from all vectors already chosen. The surprising result is that C is a linear code. Codes found in this fashion are called greedy codes. Greedy codes are generated by applying greedy algorithm on vectors arranged in B-ordering.

Codes constructed via this algorithm have been optimal or near-optimal, with dimensions either the highest possible for a given length and minimum distance or within one of the highest possible. In fact such codes satisfy Varshamove – Gilbert bound, which is the best known lower bound. W Chen[5],[6] used greedy algorithm to find the weight hierarchies of binary linear codes of dimension 4.

3.1 G-VALUES

In this section we associate with each vector in a B-ordering a non negative integer, called a g-value [2]. These nonnegative integers may be written as binary vectors representing the integers in the usual way. g-values may be assigned to the vectors in the following recursive manner. Each vector is considered in order and the first vector, which is the zero vector, is assigned 0 as its g-value; then if ***v*** is a vector under consideration and G is the set of g-values of all previous vectors which are at distance less than d from ***v***, then the g-value of ***v*** is the least nonnegative integer which is not an element of G. The set of all vectors having g-value 0, is infect the greedy code itself. Thus the assignment of g-values is a generalization of the basic greedy algorithm for generating codes.

The g-values may also be assigned in order, as opposed to assigning to each vector in order a g-value. One may first select all vectors with g-value 0 (the code); then select all vectors with g-value 1, etc. rather than assigning the first a g-value, then the second vector its g-value, etc. Either method produces the same assignment of g-values.

V.Pless[2] showed that $g: R^n \longrightarrow S^m$ is a homomorphism and its kernel is a binary greedy code.

Theorem: Let $g: R^n \longrightarrow S^m$ be a homomorphism from ring R^n to ring S^m and $s \in g(R^n)$. Then $g^{-1}(s)$ equals the coset ker (g)+ r, where r is any given element of $g^{-1}(s)$.

Proof:

Let $s \in g(R^n)$, and choose any $r \in g^{-1}(s)$ (which is non-empty by assumption). We must show that the sets ker(g) + r and $g^{-1}(s)$ are equal. Choose an arbitrary element $a + r \in \ker(g) + r$,
where $a \in \ker(g)$.
Then $g(a + r) = g(a) + g(r) = 0 + g(r) = s$.
Thus, $a + r \in g^{-1}(s)$, as claimed.

Conversely, choose $t \in g^{-1}(s)$. Then consider t + r;
$g(t + r) = g(t) + g(r) = s + s = 0$,
and so $t + r \in \ker(g)$. But then $t = (t + r) + r \in \ker(g) + r$, as required. □

By the definition of function pre-images of distinct elements are disjoint from one another, so the set of cosets of kernel decomposes the domain ring into set of pair wise disjoint subsets. In other words the set of cosets of kernel partitions the ring. The unique one of these sets containing '0' is an ideal since an ideal has to contain zero it is clear that only one of the cosets can be an ideal. One can think of the ideal itself as a coset ker g+0, implies binary greedy code is an ideal of a ring consists of all strings of length n.

Instead of calculating the g-values of all words in R^n one only needs to know kernel of g , then we capture all the pre- images of a ring homomorphism by additively translating the kernel.

Example:

The assignment of g-values to a B-ordering of binary vectors. The length n is 5, and the chosen distance d is 3. The basis elements are is bold face.

R^5 ={00000,00001,00011,00010,00111,00110,0010,00101,01111,01110,01100,01101,01000,01001,01011,01010,11111,11110,11100,11101,11000,11001,11011,11010,10000,10001,10011,10010,10110,10100,10101}

Let $g: R^5 \longrightarrow S^3$ as defined

$\ker(g)=\{\boldsymbol{v} \mid g(\boldsymbol{v})=0 \ \forall \ \boldsymbol{v} \in R^5\}$
$=\{00000,00111,11110,11001\}=G_0$

pick any word $\boldsymbol{w} \in R^5 \setminus G_0$ let ***w***=00110,calculate

ker(g)+00110={00110,00001,11000,11111}= G_1

pick another word say 00011= $\boldsymbol{w} \in R^5 \setminus G_0 \cup G_1$ and calculate,

ker (g)+00011={00011,00100,11101,11010}=G_2

Continue in this way until the list exhausted

g-values calculated by this method may differ from the g-values calculated by V. Pless[2] but this difference could not effect the decoding scheme.

3.1.1 Hamming greedy codes

We can produce perfect greedy codes with the parameters of Hamming codes via greedy algorithm. Let all binary n-tupples v_n be in their B-ordering and choose a distance d. We construct a set C consisting of the vectors with g-value zero; C is actually a linear code (n, k, d). With d=3,n=2r-1, r>2,k=2r-1-r

Hamming (2r-1, 2r-1 -r,3) codes are obtained.
For example let r =3 then n = 7; k = 4; d = 3, (7,4,3) Hamming code is produced.

3.1.2 Hamming greedy codes (7,4,3)

B={0000001,0000011,0000111,0001111,00111, 0111111,1111111}

Consider B as the basis for B-ordering.

By definition of greedy codes C consists of all words with g-value 0.

C={0000000, 0000111, 0011110, 0011001, 0110011, 0110100, 0101101, 0101010, 1111111, 1111000, 1100001, 1100110, 1001100, 1001011, 1010010, 1010101}

Table 1

g-value / S.No.	0	1	2	3	4	5	6	7
1	0000000	0000001	0000011	0000010	0001111	0001110	0001100	0001101
2	0000111	0000110	0000100	0000101	0001000	0001001	0001011	0001010
3	0011110	0011111	0011101	0011100	0010001	0010000	0010010	0010011
4	0011001	0011000	0011010	0011011	0010110	0010111	0010101	0010100
5	0110011	0110010	0110000	0110001	0111100	0111101	0111111	0111110
6	0110100	0110101	0110111	0110110	0111011	0111010	0111000	0111001
7	0101101	0101100	0101110	0101111	0100010	0100011	0100001	0100000
8	0101010	0101011	0101001	0101000	0100101	0100100	0100110	0100111
9	1111111	1111110	1111100	1111101	1110000	1110001	1110011	1110010
10	1111000	1111001	1111011	1111010	1110111	1110110	1110100	1110101
11	1100001	1100000	1100010	1100010	1101110	1101111	1101101	1101100
12	1100110	1100111	1100101	1100100	1101001	1101000	1101010	1101011
13	1001100	1001101	1001111	1001110	1000011	1000010	1000000	1000001
14	1001011	1001010	1001000	1001001	1000100	1000101	1000111	1000110
15	1010010	1010011	1010001	1010000	1011101	1011100	1011110	1011111
16	1010101	1010100	1010110	1010111	1011010	1011011	1011001	1011000

3.2 Properties of g-values

For different values of n, k and d. database suggests the following very useful properties of g-values. For any (n, k, d) linear code C. $\boldsymbol{u}, \boldsymbol{v} \in Z_2^n$

i) If a word $\boldsymbol{u}$ has g-value i, then G_i consists of all words with g-value i Each word $\boldsymbol{u}$ with g-value 3 is in G_3

G_3 = {0000010, 0000101, 0011100, 0011011, 0110001, 0110110, 0101111, 0101000, 1111101, 1111010, 1100011, 1100100, 1001110, 1001001, 1010000, 101011}

ii) If $\boldsymbol{u} \oplus \boldsymbol{v}$ is in C, then $\boldsymbol{u}$ and $\boldsymbol{v}$ have the same g-value

Let $\boldsymbol{u}$=1010001 and $\boldsymbol{v}$ = 1010110, $\boldsymbol{u} \oplus \boldsymbol{v}$ = 0000111 $\in$ C so each $\boldsymbol{u}$ and $\boldsymbol{v}$ have the same g-value i. e 2.

iii) If $\boldsymbol{u} \oplus \boldsymbol{v}$ is not in C then $\boldsymbol{u}$ and $\boldsymbol{v}$ has different g-value.

$\boldsymbol{u}$ = 1110111, $\boldsymbol{v}$ =1110010 since $\boldsymbol{u} \oplus \boldsymbol{v}$ = 0000101 $\notin$ C so $\boldsymbol{u}$ has g-value 4 and $\boldsymbol{v}$ has g-value 7.

iv) No. of words in each G_i is the same as number of words in C.

i. e., $|G_i| = |C|$.

$|G_i| = 16 \quad \forall$ i =0,1,2,3,4,5,6,7.

$|C| = 16 \quad |G_i| = |C|=16$

If C has dimension k. then there are exactly 2^{n-k} different g-values.

In case of (7, 4, 3)-Hamming code, dimension k=4, length n=7, then

Number of different g-values = $2^{n-k} = 2^{7-4} = 2^3 = 8$

v) $g(\boldsymbol{u} \oplus \boldsymbol{v}) = g(\boldsymbol{u}) + g(\boldsymbol{v})$ in binary expansion of g-values.

Let $\boldsymbol{u}$ = 1011010 $g(\boldsymbol{u})$ =4=100

$\boldsymbol{v}$ = 0111010 $g(\boldsymbol{v})$ =5 =101

$\boldsymbol{u} \oplus \boldsymbol{v}$=1100000

$g(\boldsymbol{u} \oplus \boldsymbol{v})$ = 1 =001 $g(\boldsymbol{u}) + g(\boldsymbol{v})$ =001

$g(\boldsymbol{u} \oplus \boldsymbol{v}) = g(\boldsymbol{u}) + g(\boldsymbol{v})$.

vi) $C \oplus \boldsymbol{u}$ is the set of words with g-value equal to g-value of $\boldsymbol{u}$

Let $\boldsymbol{u}$ = 0101111 from the table 4 $\boldsymbol{u}$ has g-value 3.

$C \oplus \boldsymbol{u}$ = {0101111, 0101000, 0110001, 0110110, 0011100, 0110111, 0000010, 0000101, 1010000, 1010111, 1001110, 1001001, 1101010, 1100100, 1111101, 1111010}

Each element of $\boldsymbol{c} \oplus \boldsymbol{u}$ has g-value 3.

Theorem:

Let $g : R_n \longrightarrow S_m$ be a ring homomorphism with a, b $\in R_n$.

Ker(g) + a = Ker (g) + b. Iff a + b $\in$ Ker(g).

Proof:

If Ker(g) + a = Ker(g) + b, then a = 0 + a $\in$ Ker(g) + a = Ker(g) + b

Hence f k$\in$ Ker(g) such that a = k + b $\Rightarrow$ a+b = k $\in$ Ker(g)

Conversely,

If a + b $\in$ Ker(g), then a =(a+b) +b $\in$ Ker(g) + b

Since a + Ker(g) and b + Ker(g) are either disjoint are equal.

Therefore Ker(g) + a = Ker(g) + b. □

4 G-VALUE DECODING

There are decoding schemes using cosets and syndrome decoding array (SDA). Comparison of the above stated properties of g-values with the properties of cosets and syndrome decoding array invites us to use g-values for decoding. Let "C" be a linear code and assume that code word $\boldsymbol{v} \in$ C is transmitted and word $\boldsymbol{w}$ is received. Let $\boldsymbol{u}=\boldsymbol{v}+\boldsymbol{w}$, $\boldsymbol{u}$ gives us information about the bits received in error. Since $\boldsymbol{u}+\boldsymbol{w}=\boldsymbol{v} \in$C , by above theorem the error pattern $\boldsymbol{u}$ and the received word $\boldsymbol{w}$ has same g-value also keeping in view the general assumption that errors are as small as possible. Now we can decode a received word with following simple algorithm.

Let $\boldsymbol{u} \in Z_2^n$ be a received word find its g-value, let it is i. Build the set G_i of all words with g-value i. Trace a word $\boldsymbol{w}$ of least weight in G_i . Then $\boldsymbol{u} \oplus \boldsymbol{w}$ is the code transmitted.

Example

Hamming (7, 4, 3) is a single error correcting code.

Let $\boldsymbol{u}$ = 1110101 be received. From table 1, $\boldsymbol{u}$ has g-value 7.

From table 1

G_7 ={0001101,0001010,0010011,0010100,0111110, 0111001,0100000,0100111,1110010,1110101,1101100,1101011,1000001,1000110, 1011111,1011000}.

Word $\boldsymbol{w}$ = 0100000 $\in G_7$ is of least weight.

$\boldsymbol{u} \oplus \boldsymbol{w}$ = 1110101 $\oplus$ 0100000 = 1011000 is the code transmitted.

5 REFERENCES

[1] E.R.Berlekamp and J.H.Conway , Winning ways for your mathematical plays. Academic Press, 1982.

[2] R.A Brualdi and V. Pless, "Greedy Codes", JCT (A) 64 (1993), 10-30.

[3] J.H. Conway, "Integral Lexicographic Codes", Discrete Mathematics 83 (1990) 219-235.

[4] L.Monroe, "Binary Greedy Codes", Congressus Numerantium, vol. 104(1994), 49-63.

[5] W.Chen and T. Kløve, "On the second greedy weight for linear codes of dimension 3" Discrete Math. vol. 241, pp. 171-187, 2001

[6] W. Chen and T. Kløve, Weight hierarchies of binary linear codes of dimension 4, Discrete Mathematics 238 (2001), 27-34.

[7] Ari Trachtenberg, Alexander Vardy, Lexicographic codes, "Lexicographic Codes" Proc.\ 31st Annual Conference on Information Sciences and Systems, 1997.

[8] W.C. Huffman and V. Pless "Fundamentals of Error- Correcting Codes" Cambridge University Press 2003.