Mathematical Problems of Computer Science 51, 7--20, 2019.

UDC 519.712.6

Complexity of Error-Correcting Code Based on Nearest Neighbor Search Algorithm[1]

Levon H. Aslanyan and Hayk E. Danoyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: lasl@sci.am, hed@ipia.sci.am

Abstract

The Nearest Neighbor search algorithm considered in this paper is well known (Elias algorithm). It uses error-correcting codes and constructs appropriate hash-coding schemas. These schemas preprocess the data in the form of lists. Each list is contained in some sphere, centered at a code-word. The algorithm is considered for the cases of perfect codes, so the spheres and, consequently, the lists do not intersect. As such codes exist for the limited set of parameters, the algorithm is considered for some other generalizations of perfect codes, and then the same data point may be contained in different lists. A formula of time complexity of the algorithm is obtained for these cases, using coset weight structures of the mentioned codes.

Keywords: NN search, Best match, Hash-coding schema, Perfect Codes, Uniformly Packed codes, Quasi-perfect codes.

1. Introduction

Let $E = \{0,1\}$. Consider Cartesian degree E^n, which is known as the set of vertexes of n-dimensional unit cube. For any $x, y \in E^n$ denote by $d(x,y)$ the Hamming distance between vectors x and y. For an arbitrary $x \in E^n$ denote by $S_r^n(x)$ the sphere of radius r, centred at x, i.e., $S_r^n(x) = \{y/y \in E^n, d(x,y) \leq r\}$ and by $O_r^n(x)$ denote the shell of radius r, centred at x, i.e., $O_r^n(x) = \{y/y \in E^n, d(x,y) = r\}$. We will denote by $car(x)$ the carrier of vector $x =$

[1] Partially supported by grants № 18RF-144, and № 18T-1B407 of the Science Committee of the Ministry of Education and Science of Armenia

$(x_1, \dots, x_n)$ then $car(x) = \{i / x_i = 1, i = 1, \dots, n\}$. Denote by $w(x)$ the weight of vector x, i.e., $w(x) = \sum_{i=1}^{n} x_i$.

Let us have a subset $F \subseteq E^n$ and a vector $x \in E^n$. Let us consider the problem of finding the set of nearest neighbors of x from F. More precisely it is required to find the set $F_x = \{y \in F / d(x, y) = d(x, F)\}$. Hash coding schemaes are considered [1,2] for preprocessing the data. A brief description of such schemes is brought below.

Hash function is defined as a function $h: E^n \rightarrow V$, where $V = \{v_1, \dots, v_N\}$ is a finite set of N elements [1]. Cases are usually considered, when $V = E^k$, $k \leq n$. The subset F is represented as a union of N disjoint sets (lists). Denote by B_i the set $\{x \in E^n / h(x) = v_i\}$. The i-th list L_i stores those vectors belonging to F, which have the same hash value, i.e., $L_i = \{x \in F / h(x) = v_i\}$ or $L_i = B_i \cap F$, $i = 1, \dots, N$. Hash coding schemae is called balanced if $|B_i| = 2^n / N$.

The Elias algorithm [2] considers blocks B_i ordering them by their distances at vector x. Mention that we must have an efficient method to find all blocks $B_{j_1}, B_{j_2}, \dots, B_{j_{s(j)}}$ located at distance j from x if such blocks exist. After the step of ordering, the algorithm examines the lists $L_{j_1}, L_{j_2}, \dots, L_{j_{s(j)}}$ one after another by increasing j_t. Let the best match distance be denoted by δ (also the current value of the best match distance in the algorithm). Due to $F \neq \emptyset$ initialization of δ will happen in some step. Now, if the current values obey $\delta < j$ algorithm stopes the work. All blocks with higher distances than δ at x do not need to be examined. In the reminder case $\delta \geq j$, examining the nonempty list L_{j_t} the algorithm can change the best match distance δ, also refreshing the current best match set, or the δ will remain unchanged and the current best match set will be updated. The pseudocode of the algorithm is brought below:

```
Elias Algorithm  // $n$ is the word length, $N$ is the number of blocks
input $x, F$,  // $F \neq \emptyset$
integer $\delta = \infty$,  // the current best match distance
set $S = \emptyset$,  // $S$-is the current set of vectors of $F$ located at distance $\delta$ from $x$
integer $j = -1$, // current distance of blocks under consideration from $x$
while($j < \delta$)
      {
        j++,
        if($s(j) \neq 0$)  // $s(j)$ is the number of blocks located at distance $j$ from $x$
          for(integer i=0, i<s(j), i++)
             {
               if($L_{j_i} \neq \emptyset$)  // start examine the list L$_{j_i}$, i-th list with j distanse block
               if($\delta \leq d(x, L_{j_i})$)  // δ is  unchanged
               $S = S \cup (O_\delta^n(x) \cap L_{j_i})$  // $O_\delta^n(x)$ is the $\delta$ neighborhood of $x$
               else
                    {
                      $S = O_\delta^n(x) \cap L_{j_i}$,  // δ is changed
                      $\delta = d(x, L_{j_i})$
                    }
             }
      }
return $S$,  // $S = F_x$, $\delta = d(x, F)$
```

Fig. 1: Pseudocode of Elias Algorithm.

By the complexity of the algorithm we mean the average number of examined lists over all files and queries, supposing that
a) each vector $x \in E^n$ equally likely can be requested.
b) each vector $z \in E^n$ independently appears in F with the same probability p. This gives probabilistic distribution over the set of subsets of E^n.

It is known [2, 3] that the algorithm is optimal, when the blocks are isoperimetric sets, a particular case of which is a sphere.

2. Coset Weight Distribution of Uniformly Packed Codes

A nonempty subset C of E^n we call a code [4]. The code C will be called linear if C is a linear subspase of E^n. Denote by d_C the minimum distance of code C i.e. $d_C = \min_{\substack{c_1,c_2\in C\\ c_1\neq c_2}} d(c_1, c_2)$. The packing radius [4] of C is called the following nonnegative integer: $r_C = [(d_C - 1)/2]$. Denote by R_C the covering radius of the code C, i.e., $R_C = \max_{x\in E^n} \min_{c\in C} d(x, c)$. In the sequel, when it doesn't cause a confusion, we use notations d, r and R instead of d_C, r_C and R_C, respectively. We say that we have a code $C[n, k, d]R$ if the code C is linear, have dimension k, codes length n, minimum distance d and covering radius R. When the code is nonlinear (or it is not known whether the code is linear or not), we use the notation $C(n, M, d)R$ instead, where $M = |C|$. We also use this for linear codes as the second alternative notation. Recall that the code C is called perfect [4], if $r = R$. It is known [4, 5] that in binary space nontrivial perfect codes can have only the following two parameter sets:
(I) $(2^m - 1, 2^{2^m - m - 1}, 3)1$,
(II) [23,11,7]3.
Here (I) corresponds to the parameters of Hemming codes and (II) refers to the case of Golay code.

For $x \in E^n$ the coset of linear code C is called the set $x + C = \{x + c/c \in C\}$. As it is known [4], two different cosets do not intersect, and their union covers the space E^n. We denote by G_C the generator matrix of the linear code $C[n, k]$, which rows forming a basis of code C. Let us denote by H_C the parity check matrix of linear code C. Recall that H_C is $(n - k) \times k$ matrix and for H_C holds the relation $c \in C \Leftrightarrow H_C c^T = 0$. Later, when it is clear which code we mean, we will use notations H and G instead of H_C and $G_{C,}$ respectively. For $x \in E^n$ denote by $A_i(x)$ the number of codewords of C located at distance i from x. The nonnegative integers $A_0^C, A_1^C, \ldots, A_n^C$, where $A_i^C = |\{c \in C/w(c) = i\}|$ are called weight spectra of code C. Let us denote by $W_C(x, y)$ the weight enumerator of code C: $W_C(x, y) = \sum_{i=0}^n A_i^C x^{n-i} y^i$. We can consider weight enumerators depending only on one variable, i.e., $W_C(x) = \sum_{i=0}^n A_i^C x^i$. Denote by $K_j^n(i)$ the Kravchouk polynomial of degree j [4] i. e. $K_j^n(i) = \sum_{l=0}^j (-1)^j \binom{n-i}{j-l}\binom{i}{l}$.

A code C will be called quasi-perfect if $R = r + 1$ [4], [6]. Many families of quasi perfect codes are known for the covering radius ≤ 4 [6], [9-13] but the general problem of existence of quasi-perfect codes by the given parameters isn't completely solved yet [6]. A particular class of quasi-perfect codes is the class of uniformly packed codes. A code C will be called uniformly packed [8] if there are numbers $a_1, \ldots, a_{R(C)}$ such that for $x \in E^n$ takes place $\sum_{i=0}^{R(C)} \alpha_i A_i(x) = 1$.

Theorem 1: *Let C be a uniformly packed code with parameters $a_0 a_1, \ldots, a_R$. Then the polynomial $L_C(x) = \sum_{i=0}^R \alpha_i K_i^n(x)$ has R distinct integer roots between* 0 *and* n.

Let us denote those roots by $\xi_1, \dots, \xi_R$. Mention that if C is a uniformly packed code containing zero vector then there exists a uniformly packed code with the same parameters and minimum weight β, where $0 \leq \beta \leq R$, which we will denote by C_β.
From Theorem 1 and its proof [8] follows:

Theorem 2: [8]. *For the weight function of the uniformly packed code C_β the following equality takes place:*

$$W_{C_\beta}(x) = \frac{(1+x)^n}{L(0)} + \sum_{i=1}^{R} B_{\xi_i}^{\beta}(1+x)^{n-\xi_i}(1-x)^{\xi_i}. \tag{1}$$

In (1) $B_{\xi_i}^{\beta}$-s are coefficients of the representation of polynomial $W_{C_\beta}(x)$ in basis $(1+x)^{n-i}(1-x)^i$, $i = 0, \dots, n$ (recall that the set of polynomials of degree $\leq n$ forms a linear space of dimension $n+1$). $B_{\xi_i}^{\beta}$-s can be calculated from (1) by equalizing the corresponding coefficients in left and right sides and assuming that we know the first R coefficients of $W_{C_\beta}(x)$. So, to find the coefficients $B_{\xi_i}^{\beta}$, we must solve the corresponding linear system of R equations with R variables.

From Theorem 2 follows:

$$A_j^{C_\beta} = \frac{\binom{n}{j}}{\sum_{i=0}^{R} a_i \binom{n}{i}} + \sum_{i=1}^{R}\left(B_{\xi_i}^{\beta} K_j^n(\xi_i)\right). \tag{2}$$

3. Hamming Code, Extended Hamming Code, Golay Code and Two Error-Correcting Primitive BCH Codes

Denote by $\mathcal{H}_m$ the Hamming code of length $2^m - 1$. As we know [4], $\mathcal{H}$ is $[2^m - 1, 2^m - m - 1, 3]1$ perfect code. The parity check matrix of $\mathcal{H}$ is as follows:

$$H_{\mathcal{H}} = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & \ddots & 1 \\ 1 & 0 & \cdots & 1 \end{pmatrix},$$

i.e., columns of $H_{\mathcal{H}}$ are all nonzero vectors of length m. As it is known [4], perfect codes have $R+1$ different types of coset, i.e., all cosets with the same minimum weight have the same weight distribution.

Now let H_{11} be the binary Hadamard matrix of Paley type [4], i.e.,

$$H_{11} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Let I_{11} be the identity matrix of size 11×11. The generator matrix of extended Golay code [4] is as follows:

$$G(\hat{\mathcal{G}}) = \begin{pmatrix} 1_{11}^T & I_{11} & 0_{11}^T & H_{11} \\ 0 & 0_{11} & 1 & 1_{11} \end{pmatrix},$$

where by 0_m and 1_m are defined the vectors of length m consisting of all 0's and 1's, respectively.

The Golay code $\mathcal{G}$ is obtained by deleting the last coordinate from every codeword of $\mathcal{G}$. As it is known, [4] $\mathcal{G}$ is a perfect three error correcting $[23,12,7]3$ code, therefore we can consider it as a uniformly packed code with parameters $a_0 = a_1 = a_2 = a_3 = 1$. Roots of $L_{\mathcal{G}}(x)$ are $\xi_1 = 8$, $\xi_2 = 12$ and $\xi_3 = 16$.

The extended Hamming code $\widehat{\mathcal{H}}_m$ is a $[2^m, 2^m - m - 1,4]2$ code [4], which is obtained from Hamming code by adding the parity check bit, i.e., the parity check matrix of extended Hamming code is as follows:

$$\mathrm{H}_{\widehat{\mathcal{H}}_m} = \begin{pmatrix} 1 & \cdots & 1 \\ & & 1 \\ & \mathrm{H}_{\mathcal{H}} & \vdots \\ & & 1 \end{pmatrix}.$$

As it is known, the extended Hamming code is a uniformly packed code [6] and has four types of coset weight distribution.

Let us denote a finite field of q elements (where q is a power of a prime number) by F_q. We will consider finite fields with characteristic 2. Denote by α the primitive element of the field F_q.Consider the set of formal polynomials $F_q[x]$ with coefficients from the field F_q. As it is known [4], the factor ring $R[x] = F_q[x]/(x^n - 1)$ is a ring of principal ideals, i.e., each ideal in $R[x]$ is principal. An $[n,k]$ code C will be called a cyclic code if C is linear, and if from $c = (c_1, c_2, \dots, c_n) \in C$, it follows that $(c_n, c_1, \dots, c_{n-1}) \in C$. We can correspond the polynomial $c_1 + c_2x + \cdots + c_nx^{n-1}$ to each vector $(c_1, c_2, \dots, c_n)$, so we can consider a code as the subset of $R[x]$. It is known [4], that each cyclic code is an ideal of $R[x]$, i.e., there is a unique polynomial $g(x)$ such that $\forall c(x) \in C \; \exists f(x) \; c(x) = f(x)g(x)$, where multiplication is taken in $R[x]$.

Two error correcting BCH codes (denoted by $\mathfrak{B}_m$) are defined as cyclic codes for lengths $n = 2^m - 1$ [4,5] with a generator polynomial:

$$g(x) = LCM\{M_\alpha(x), M_{\alpha^3}(x)\},$$

where by $M_{\alpha^i}(x)$ is denoted the minimal polynomial of the element α^i. These codes have the dimension $2^m - 2m - 1$ and minimum distance equal to 5 [4]. It is known that two error correcting BCH codes are quasi-perfect codes [4,13,14]. The weight distribution of BCH codes was calculated in [4, 13,14]. For odd m two error correcting BCH codes are also uniformly packed [8] with parameters $a_0 = a_1 = 1$, $a_2 = a_3 = \frac{6}{n-1}$. Roots of $L_{\mathfrak{B}}(x)$ are $\xi_1 = \frac{n+1}{2} - \sqrt{\frac{n+1}{2}}$, $\xi_2 = \frac{n+1}{2}$ and $\xi_3 = \frac{n+1}{2} + \sqrt{\frac{n+1}{2}}$. It is known, that there are four distinct coset weight distributions.

For even m two error correcting BCH codes are not uniformly packed. It is proved that there are eight distinct coset weight distributions in this case, which are brought in [14].

4. Complexity of the Algorithm

Suppose we have an $[n,k]$ code C with covering radius R and $C = \{c_1, c_2, \dots, c_{2^k}\}$. We define a hash function $h: E^n \longrightarrow C$, associated to the code C in the following way:

$$h_C(x) = \left\{c_i / d(x, c_i) = \min_{c \in C}\{d(x,c)\}\right\}. \tag{3}$$

As it follows from (3), $h_C(x)$ could be a multivalued function because the blocks B_i are spheres of radius R, and they can intersect (recall that $B_i = \{x \in E^n / h_C(x) = c_i\}$, $i \in \{1, \dots, 2^k\}$). When the code C is perfect, the mentioned blocks do not intersect, and their union covers the unit cube. The formula below for complexity of algorithm is brought for the case corresponding to Hamming code. We also consider hash functions associated to codes in some sense "near" the perfect codes. Such property also has the so called quasi-perfect codes. Indeed, the algorithm is proposed for balanced hash coding schemas where different blocks B_i do not intersect; we also consider the algorithm for the case of intersecting blocks. In this case, when blocks intersect, we create the list in a similar way. Repeated elements bring some redundancy (in terms of memory).

To obtain a formula of complexity of the algorithm, for $x \in E^n$ let us consider Fig. 2. In Fig. 2 $F_1, F_2, \dots, F_{2^{2^n}}$ are all subsets of vertexes of unit cube and each F_i could be generated with the corresponding probabilityp_i.

	x	p_1	p_2		$p_{2^{2^n}}$	probability
		F_1	F_2	⋯	$F_{2^{2^n}}$	subset
	B_1	a^x_{11}	a^x_{12}	⋯	$a^x_{12^{2^n}}$	
blocks	B_2	a^x_{21}	a^x_{22}	⋯	$a^x_{22^{2^n}}$	
	⋮	⋮	⋮	⋯	⋮	
	B_{2^k}	$a^x_{2^k1}$	$a^x_{2^k2}$	⋯	$a^x_{2^k2^{2^n}}$	

Fig. 2.

We will use the values a^x_{ij} putting them in the cells corresponding to block B_i and subset F_j, where

$$a^x_{ij} = \begin{cases} 1, if\ B_i is\ considered\ in\ case\ of\ set\ F_i\ and\ vertex\ x, \\ 0\ otherwise. \end{cases}$$

As we mentioned, the complexity of the algorithm will be represented as

$$\alpha(h_C) = \frac{1}{2^n} \sum_{x \in E^n} \sum_{1 \le i \le 2^k} \sum_{1 \le j \le 2^{2^n}} p_j a_{ij}^x$$

Let us denote $\Phi_x(B_i) = \sum_{1 \le j \le 2^{2^n}} p_j a_{ij}^x$. As we can see, $\Phi_x(B_i)$ is the probability that the block B_i will be considered by the algorithm when the vector x is requested. Then

$$\alpha(h_C) = \frac{1}{2^n} \sum_{x \in E^n} \sum_{1 \le i \le 2^k} \Phi_x(B_i).$$

It is easy to understand that for a fixed query x the block B_i will be examined if the sphere $S_{d(x,B_i)-1}^n$ does not contain any vector belonging to F. In that case, all blocks B_l such that $d(x, B_l) \le d(x, B_i) - 1$, will be examined. Let j vary over all possible distances between vector x and blocks B_i. Denote by $T_x(j)$ the number of blocks located at distance $\le j$ from vector x, then

$$\alpha(h_C) = \frac{1}{2^n} \sum_{x \in E^n} \sum_{0 \le j \le n} T_x(j) V(j), \tag{4}$$

where V(j) denotes the probability that the nearest vector in F is located at distance j from x. Recall that [2]

$$V(j) = \left(1 - (1-p)^{\binom{n}{j}}\right)(1-p)^{\sum_{l=0}^{j-1}\binom{n}{l}}.$$

As $d(x, C_i) = w(x + c_i)$, then the number of vectors located at distance i is equal to A_i^{x+C}. The sphere with centre c_i and radius R will be located at a distance $\le j$ from vector x if and only if $d(x, c_i) \le j + R$. Therefore

$$T_x(j) = \sum_{i=0}^{j+R} A_i^{x+C}. \tag{5}$$

We consider that $A_i^{x+C} = 0$ when $i > n$.

Taking into account formulas (4) and (5) and the coset weight structures for considered codes, we may formulate the following:

Proposition 1: *The complexity of the algorithm for the hash function defined by* $[2^m - 1, 2^m - m - 1, 3]1$ *Hamming code* $\mathcal{H}_m$ *is:*

$$\alpha\left(h_{\mathcal{H}_m}\right) = \frac{1}{2^m} \sum_{0 \le j \le 2^m - 1} V(j) \left(\sum_{i=0}^{j+1} \left(A_i^{\mathcal{H}_m} + (2^m - 1) A_i^{e_1 + \mathcal{H}_m}\right) \right), \tag{6}$$

where by e_i *is denoted any fixed vector of weight i.*

Proposition 2: *For the Golay code* $\mathcal{G}$ *the complexity of the algorithm is:*

$$\alpha\left(h_{\mathcal{G}}\right) = \sum_{0 \le j \le 23} V(j) \sum_{i=0}^{j+3} \left(\frac{1}{2^{11}} A_i^{\mathcal{G}} + \frac{23}{2^{11}} A_i^{e_1 + \mathcal{G}} + \frac{253}{2^{11}} A_i^{e_2 + \mathcal{G}} + \frac{5819}{2^{11}} A_i^{e_3 + \mathcal{G}} \right). \tag{7}$$

Proposition 3: *For the code $\widehat{\mathcal{H}}_m$ the complexity of the algorithm is:*

$$\alpha\left(h_{\widehat{\mathcal{H}}_m}\right) = \sum_{0\le j\le 2^m} V(j)\left(\sum_{i=0}^{j+2}\left(\frac{1}{2^{m+1}}A_i^{\widehat{\mathcal{H}}_m} + \frac{1}{2}A_i^{\mathrm{e}_1+\widehat{\mathcal{H}}_m} + \frac{2^m-1}{2^{m+1}}A_i^{u+\widehat{\mathcal{H}}_m}\right)\right), \tag{8}$$

where by u_i is denoted any fixed vector of weight 2, the first coordinate of which is equal to 1.

Proposition 4: *For two error correcting BCH code $\mathfrak{B}_m$ of length 2^m-1 for odd m the complexity of the algorithm is:*

$$\alpha\left(h_{\mathfrak{B}_m}\right) = \sum_{0\le j\le 2^m-1} V(j)\sum_{i=0}^{j+3}\left(\frac{1}{2^{2m}}A_i^{\mathfrak{B}_m} + \frac{2^m-1}{2^{2m}}A_i^{\mathrm{e}_1+\mathfrak{B}_m} + \right. \\ \left. + \frac{(2^m-1)(2^{m-1}-1)}{2^{2m}}A_i^{\mathrm{e}_2+\mathfrak{B}_m} + \frac{2^{2m-1}+2^{m-1}-1}{2^{2m}}A_i^{\mathrm{e}_3+\mathfrak{B}_m}\right). \tag{9}$$

5. Numeric Results

Even having formulas, it is hard to imagine the practical complexities of things. To compare the results with those in [2], we provide numeric results for propositions 1 to 4 to demonstrate the complexity of the algorithm for each case of hash-coding schema. Table 1 demonstrates the formula (6) in cases of Hamming code of length 2^m-1 for $m=4$. Table 2 demonstrates the formula (7) in case of Golay code. Tables 3 and 4 demonstrate the formula (8) in cases of extended Hamming code of length 2^m-1 for $m=4$ and $m=5$. Table 5 demonstrates the formula (9) in case of BCH codes of length 2^m-1 for $m=5$.

Table 1: Complexity of the algorithm in case of Hamming code of length 15.

Subset generation probability p	Average number of considered blocks	The percentage relation of considered elements and cardinality of subset
1	*1*	*0.05*
2^{-1}	*4.28*	*0.21*
2^{-2}	*6.2*	*0.3*
2^{-3}	*10.1*	*0.49*
2^{-4}	*17.31*	*0.85*
2^{-5}	*26.3*	*1.28*
2^{-6}	*42.27*	*2.06*
2^{-7}	*67.67*	*3.3*
2^{-8}	*107.23*	*5.24*
2^{-9}	*170.38*	*8.32*
2^{-10}	*268.45*	*13.11*
2^{-11}	*418.26*	*20.42*
2^{-12}	*638.09*	*31.16*
2^{-13}	*901.17*	*44.0*
2^{-14}	*1001.82*	*48.92*
2^{-15}	*823.24*	*40.2*

Table 2: Complexity of the algorithm in case of Golay code of length 23

Subset generation probability p	Average number of considered blocks	The percentage relation of considered elements and cardinality of subset
1	*1*	*0.02*
2^{-1}	*3.16*	*0.08*
2^{-2}	*4.26*	*0.10*
2^{-3}	*5.45*	*0.13*
2^{-4}	*8.54*	*0.21*
2^{-5}	*12.86*	*0.31*
2^{-6}	*17.14*	*0.42*
2^{-7}	*24.51*	*0.6*
2^{-8}	*36.97*	*0.90*
2^{-9}	*51.85*	*1.27*
2^{-10}	*75.16*	*1.83*
2^{-11}	*109.80*	*2.68*
2^{-12}	*157.04*	*3.83*
2^{-13}	*227.57*	*5.55*
2^{-14}	*325.28*	*7.94*
2^{-15}	*463.76*	*11.32*
2^{-16}	*654.12*	*15.97*
2^{-17}	*911.53*	*22.25*
2^{-18}	*1249.29*	*30.50*
2^{-19}	*1674.63*	*40.88*
2^{-20}	*2176.78*	*53.14*

Table 3: The case of extended Hamming code of length $n = 16$.

Subset generation probability p	Average number of considered blocks	The percentage relation of considered elements and cardinality of subset
1	*4.28*	*0.89*
2^{-1}	*13.03*	*2.72*
2^{-2}	*17.83*	*3.72*
2^{-3}	*25.47*	*5.32*
2^{-4}	*39.69*	*8.29*
2^{-5}	*56.14*	*11.73*
2^{-6}	*80.80*	*16.89*
2^{-7}	*119.08*	*24.89*
2^{-8}	*171.14*	*35.77*
2^{-9}	*247.85*	*51.81*
2^{-10}	*354.80*	*74.17*
2^{-11}	*502.63*	*105.07*

2^{-12}	699.58	146.24
2^{-13}	947.74	198.12
2^{-14}	1196.9	250.20
2^{-15}	1235.31	258.23
2^{-16}	977.51	204.34

Table 4: The case of extended Hamming code of length n=32.

Subset generation probability p	Average number of considered blocks	The percentage relation of considered elements and cardinality of subset
1	8.26	0.0001
2^{-1}	47.01	0.0005
2^{-2}	66.43	0.0008
2^{-3}	82.93	0.001
2^{-4}	147.70	0.001
2^{-5}	280.41	0.003
2^{-6}	419.46	0.005
2^{-7}	568.55	0.007
2^{-8}	976.12	0.012
2^{-9}	1731.29	0.02
2^{-10}	2572.65	0.03
2^{-11}	4038.7	0.04
2^{-12}	7117.07	0.08
2^{-13}	11173.6	0.13
2^{-14}	18011.1	0.22
2^{-15}	30662.2	0.37
2^{-16}	48773	0.6
2^{-17}	81535.8	1.00
2^{-18}	133110	1.63
2^{-19}	218976	2.69
2^{-20}	358907	4.42
2^{-21}	587880	7.24
2^{-22}	957978	11.79
2^{-23}	$1.55722 \cdot 10^{6}$	19.17
2^{-24}	$2.511422 \cdot 10^{6}$	30.16
2^{-25}	$4.0295 \cdot 10^{6}$	49.63
2^{-26}	$6.3936 \cdot 10^{6}$	78.74
2^{-27}	$1.0009 \cdot 10^{7}$	123.27
2^{-28}	$1.53811 \cdot 10^{7}$	189.44
2^{-29}	$2.29847 \cdot 10^{7}$	283.09
2^{-30}	$3.16976 \cdot 10^{7}$	390.41
2^{-31}	$3.4639 \cdot 10^{7}$	462.64
2^{-32}	$2.82105 \cdot 10^{7}$	347.46

Table 5: The case of two error correcting BCH code of length n=31.

Subset generation probability p	Average number of considered blocks	The percentage relation of considered elements and cardinality of subset
1	*4.87*	*0.001*
2^{-1}	*20.23*	*0.004*
2^{-2}	*27.93*	*0.006*
2^{-3}	*34.07*	*0.007*
2^{-4}	*54.72*	*0.01*
2^{-5}	*94.71*	*0.02*
2^{-6}	*135.65*	*0.03*
2^{-7}	*179.04*	*0.04*
2^{-8}	*284.66*	*0.06*
2^{-9}	*463.67*	*0.10*
2^{-10}	*658.39*	*0.15*
2^{-11}	*987.25*	*0.22*
2^{-12}	*1597.11*	*0.37*
2^{-13}	*2363.61*	*0.54*
2^{-14}	*3624.62*	*0.84*
2^{-15}	*5683.15*	*1.32*
2^{-16}	*8556.05*	*1.98*
2^{-17}	*13333.3*	*3.09*
2^{-18}	*20290.5*	*4.71*
2^{-19}	*31196.3*	*7.25*
2^{-20}	*47458*	*11.03*
2^{-21}	*72165.5*	*16.77*
2^{-22}	*108815*	*25.29*
2^{-23}	*162915*	*37.87*
2^{-24}	*241404*	*56.11*
2^{-25}	*353024*	*82.06*
2^{-26}	*507518*	*117.977*
2^{-27}	*713305*	*165.813*

2^{-28}	*971293*	*225.785*
2^{-29}	$1.22855\cdot10^{6}$	*285.587*
2^{-30}	$1.26797\cdot10^{6}$	*294.749*
2^{-31}	$1.00312\cdot10^{6}$	*233.184*

6. Conclusion

The Nearest Neighbor search algorithm considered in this paper is well known (Elias algorithm). It uses hash-coding schemas for data preprocessing. These schemas partition the space into non-intersecting blocks of the same cardinality. It is known that the algorithm is optimal when these blocks are spheres. Such partitions may be obtained by error-correcting codes. The algorithm is considered for the cases of perfect codes, so the spheres and, consequently, the lists do not intersect. As such codes exist for a limited set of parameters, the algorithm is considered for some other generalizations of perfect codes, and then the same data point may be contained in different lists. A formula of time complexity of the algorithm is obtained for these cases. These formulas show the area of practical use of the algorithm: the algorithm encounters “the course of dimensionality”, i.e., as the word length grows, the algorithm turns into an exhaustive search in a file.

References

[1] D. E. Knuth, *The Art of Computer Programming*, vol. 3, Sorting and Searching, second edition, Eddison-Wesley, 1998.

[2] R. Rivest, “On the optimality of Elias’s algorithm for performing best-match searches”, *Information Processing*, pp. 678–681, 1974.

[3] L. H. Aslanyan and H. E. Danoyan, “On the optimality of the hash-coding type nearest neighbour search algorithm”, Selected works of 9th CSIT conference, pp. 1-6, 2013.

[4] F. J. Mac-Williams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* Amsterdam, The Netherlands: North-Holland, 1986.

[5] V. A. Zinoviev and V. K. Leont’ev, “The nonexistence of perfect codes over Galois fields”, *Problems of Control and Information Theory*, vol. 2, no. 2, pp. 123-132, 1973.

[6] L. A. Bassalygo, G. V. Zaitsevand V. A. Zinoviev, “Uniformly packed codes”, Problemy Peredachi Informatsii, vol. 10, no. 1, pp. 9-14, 1974.

[7] T. Baicheva, I. Bouyukliev and S. Dodunekov, “Binary and ternary linear quasi-perfect codes with small dimensions”, *IEEE Transactions of Information Theory*, vol. 54, no. 9, pp. 4335-4339, 2008.

[8] E. M. Gabidulin, A. A. Davydov and L. M. Tombak, “Linear codes with covering radius 2 and other new covering codes,” *IEEE Transactionsof Information Theory*, vol. 37, no. 1, pp. 219–224, 1991.

[9] A. A. Davydov and A. Yu. Drozhzhina-Labinskaya, “Constructions, families, and tables of binary linear covering codes,” *IEEE Transactions of Information Theory*, vol. 40, no. 4, pp. 1270–1279, Jul. 1994.

[10] T. Etzion and B. Mounits, “Quasi-perfect codes with small distance”, *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3938-3946, 2005.

[11] T. Etzion and G. Greenberg, “Constructions for perfect mixed codes and other covering codes,” *IEEE Transactions of Information.Theory*, vol. 39, no. 1, pp. 209–214, 1993.

[12] D. Gorenstein, W. Peterson and N. Zierler, “Two error-correcting bose-chaudhuri codes are quasi perfect”, *Information and Control*, no. 3, pp. 291-294, 1960.

[13] T. Kasami, S. Lin and W. Peterson, “Some results on the weight distributions of BCH codes”, *IEEE Trans. Information Theory*, vol. 12, no. 2, pp. 274-277, 1966.

[14] P. Charpin “Weight distributions of cosets of two-error-correcting BCH codes, extended or not”, *IEEE Transactions of Information.Theory*, vol. 40, no. 5, pp. 1425–1442, 1991.

Submitted 18.02.2019, accepted 15.04.2019.

Սխալ ուղղող կոդերի վրա հիմնված մոտակա հարևանների փնտրման ալգորիթմի բարդությունը

Լևոն Հ. Ասլանյան և Հայկ Է. Դանոյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail: lasl@sci.am, hed@ipia.sci.am

Ամփոփում

Հայտնի է մոտակա հարևանների փնտրման հաշ-կոդավորման տիպի Էլիասի ալգորիթմը: Ալգորիթմում կարող են կիրառվել սխալ ուղղող կոդերը՝ հաշ-կոդավորման սխեմա կառուցելու համար: Այդ սխեմաները տվյալները ներկայացնում են ցուցակների տեսքով: Կամայական ցուցակ իրենից ներկայացնում է որևէ գնդի ենթաբազմություն (Հեմինգի տարածությունում), որի կենտրոնը նշված սխալ ուղղող կոդի կոդային բառ է: Ալգորիթմը դիտարկվում է կատարյալ կոդերի համար, հետևաբար կոդային բառերի շուրջ համապատասխան շառավղով գնդերը չեն հատվում, հետևաբար նշված ցուցակները նույնպես չեն հատվի: Քանի որ կատարյալ կոդեր գոյություն ունեն պարամետրերի սպեցիֆիկ արժեքների համար, ալգորիթմը դիտարկվել է կատարյալ կոդերի այլ ընդհանրացումներով ստացվող հաշ-կոդավորման սխեմաների համար, որտեղ, սակայն, միևնույն էլեմենտը կարող է պատկանել մի քանի ցուցակների: Նշված դեպքերի համար ստացվել են ալգորիթմի բարդության բանաձևերը՝ կախված կոդի հարակից դասերի կշռային կառուցվածքներից:

Բանալի բառեր՝ մոտակա հարևանների փնտրում, լավագույն համընկնումների փնտրում, հաշ-կոդավորման սխեմա, կատարյալ կոդեր, հավասարաչափ փաթեթավորված կոդեր, քվազիկատարյալ կոդեր

Сложность алгоритма поиска ближайших соседей с кодами исправляющими ошибки

Левон А. Асланян и Айк Э. Даноян

Институт проблем информатики и автоматизации НАН РА
e-mail: lasl@sci.am, hed@ipia.sci.am

Аннотация

Известен алгоритм поиска ближайших соседей (алгоритм Элиаса). Алгоритм использует коды, исправляющие ошибки для построения схем хеш-кодирования. Эти схемы представляют данные в форме списков. Каждый список содержится в некоторой сфере, центром которого является некоторое кодовое слово. Алгоритм рассматривается для случаев совершенных кодов, поэтому указанные сферы и, следовательно, списки не пересекаются. Поскольку совершенные коды существуют для очень специфичного набора параметров, алгоритм рассматривается для некоторых обобщений совершенных кодов, когда одна и та же точка данных может содержаться в разных списках. Для указанных случаев получены формулы временной сложности алгоритма с использованием весовой структуры смежных классов кодов.

Ключевые слова: поиск ближайших соседей, поиск наилучших совпадений, схемы хеш-кодирования, совершенные коды, равномерно упакованные коды, квазисовершенные коды