On the Complexity of Generating Forney Codes

Marjan Gusev

ON THE COMPLEXITY OF GENERATING FORNEY'S CODES

Dejan Spasov, Marjan Gusev
Institute of Informatics
Faculty of Natural Sciences
Skopje, Macedonia

Abstract

Concatenated codes are code constructions made of two codes called the inner code and the outer code [1]. The outer code is usually asymptotically good code over a large alphabet F_{q^m}, like the Reed-Solomon code. If a greedy code is used as an inner code, then, following the terminology from [2], we call these codes Forney's codes. In [2], it is suggested that the best code in Wozencraft's ensemble should be used as an inner code; thus lowering the complexity on finding a good inner code. In this paper we present four greedy algorithms that can be used to produce the inner code. Some of these algorithms have lower time complexity than finding the best code in the Wozencraft's ensemble.

I. Introduction

Let F_q is a finite field of q elements and let F_q^n is n-dimensional vector space over F_q. Then a *code* C is subset of F_q^n of M elements. Elements of the code $c_i \in C$ are called *codewords*.

Let $d(x,y)$ denotes the *Hamming distance*, i.e. the number of coordinates in which two vectors x and y differ, and let $wt(x)$ denotes the (*Hamming*) *weight*, i.e. the number of nonzero coordinates of x. Then we say that the code C has *(minimum) distance* d if

$$d = \min\left\{d\left(c_i, c_j\right)\right\}, \forall c_i, c_j \in C, i \neq j. \tag{1}$$

We are interested in asymptotical behavior of linear codes. Thus a single code is of no interest to us, but an infinite family of codes C_i, $i \to \infty$, of increasing length $n_i \to \infty$. Infinite code families are more convenient to be described in terms of the *code rate* $R = \liminf_{i\to\infty}\left(\frac{k_i}{n_i}\right)$ and the *relative distance* $\delta = \liminf_{i\to\infty}\left(\frac{d_i}{n_i}\right)$.

The code C is linear if its codewords form k-dimensional linear subspace in F_q^n. We will write $[n,k,d]_q$ to denote that the code C is linear over the field F_q. For linear codes there exist k basis vectors that are kept as rows in a matrix G called the *gererator matrix*. For each linear code there is a generator matrix of type $G = [I \quad A]$ for which we say that is in *standard form*. It is well-known that for linear codes there exist a so-called *parity check matrix* H, such that $\forall c_i \in C \quad Hc_i^T = 0$. Let $G = [I \quad A]$ is the generator matrix, then $H = \left[-A^T \quad I\right]$ is the parity check matrix of the same code.

The following theorem is a fundamental result in coding theory:

Theorem 1: The code C with parameters $[n,k,?]_q$ and parity check matrix H has minimal distance d if every linear combination of $d-1$ columns of H is linearly independent, and there exist a linearly dependent combination of d columns of H.

The *covering radius* of a code is the largest possible distance between the code C and a vector from F_q^n, i.e.

$$\rho = \max_{x\in F_q^n} \min_{\forall c\in C} d(x,c). \tag{2}$$

We will use $(x \mid y)$ to denote concatenation of two strings, and x^k to denote a string of k symbols x, namely $\underbrace{x \ldots x}_{k}$.

We will use $Ball(x,d)$ to denote a *Hamming ball* with radius d and center in x,

$$Ball(x,d) = \left\{y \in F_q^n \mid d(x,y) \le d\right\}, \tag{3}$$

and $V(n,d)$ is *the volume of the ball*

$$V(n,d) = \sum_{i=0}^{d} (q-1)^i \binom{n}{i}, \tag{4}$$

The *entropy* function will be denoted with the standard notation $H(\delta)$. Using Stirling's approximation we can derive asymptotic relation between the entropy function and the volume of a ball

$$H(\delta) = \lim_{n\to\infty}\left\{\frac{\log\left(V(n,\delta n)\right)}{n}\right\}, \tag{5}$$

II. GREEDY ALGORITHMS

It is well-known that a simple greedy procedure produces an infinite code family with parameters that follow the Gilbert-Varshamov bound

$$R \geq 1 - H(\delta). \tag{6}$$

In binary case no better code family to-date is known, but, on the other hand, the greedy algorithm is considered impractical due to its exponential time complexity. In this section we give an overview of the best-known greedy algorithms.

A. Gilbert's Construction

In general, Gilbert's algorithm produces a nonlinear (n, M, d) code. Given the code length n and its minimum distance d, the algorithm will search the entire space F_q^n and greedily will add to C the first vector x that is at distance d from C, i.e. $d(x,c) \geq d, \forall c \in C$.

Theorem 2[3]: Given n and d, the time complexity of the Gilbert's algorithm in worst-case is $O\left(nq^{(1+R)n}\right)$, while the space complexity is $O\left(nq^{Rn}\right)$.

B. Varshamov's Construction

Given n and d, the Varshamov-type algorithms search over the codimension F_q^m, $m = 1, 2, \ldots$, and greedily add to the parity-check matrix H the first vector x that is NOT $(d-2)$-linear combination of columns of H. We will make a difference between two variants of the algorithm: with exponential space complexity and with polynomial space complexity.

Theorem 3[4]: Given n and d, the time complexity of the Varshamov's algorithm with polynomial space complexity in worst-case is $O\left(n^2 q^{2H(\delta)n}\right)$. The space complexity is $O\left(n^2\right)$.

Let assume that we have reserved a space of q^m bits, such that for each vector $\alpha \in F_q^m$ we have a unique bit location at the address $i[\alpha]$. Let $H(m, d-2) \subseteq F_q^m$ is the set of all vectors spanned by $d-2$ columns of H and let

$$i[\alpha] = \begin{cases} 1 & \forall \alpha \in H(m, d-2) \\ 0 & \forall \alpha \notin H(m, d-2) \end{cases}$$

Then the Varshamov's algorithm with exponential space complexity will search through the array $i[\alpha]$. The first α such that $i[\alpha] = 0$, will be added as a column to H. Then the algorithm updates the array $i[\alpha]$.

Theorem 4[4]: Given n and d, the space and time complexity of the Varshamov's algorithm with polynomial space complexity are $O\left(q^{H(\delta)n}\right)$.

C. Jenkins' Construction

The Jenkins' algorithm builds the generator matrix of a systematic code $G = [I \quad A]$. Let assume that the algorithm has already produced the generator matrix G for the code $[n, k, d]$. Then, for each $x \in F_q^m$ the algorithm forms the vector $c_{k+1} \in F_q^n$, $c_{k+1} = (10...0|x)$, and checks if all linear combinations of rows of $G_{k+1} = \begin{bmatrix} G \\ c_{k+1} \end{bmatrix}$ have weight greater than or equal to d.

Theorem 5[4]: Given n and d, the time complexity of the Jenkins' algorithm in worst-case is $O\left(n^3 q^n\right)$.

To our record this algorithm was first published by B. Jenkins in [5], so we call it the Jenkins' algorithm.

D. Lexicographic Construction

Lexicographic Construction is a variation of the Jenkins algorithm with exponential space complexity. This algorithm was first introduced in [6], then subsequently improved in [7]. Bellow it is given generalization of the algorithm over arbitrary alphabet.

Given the code $[n, k, d]$ with generator matrix G_k. Let for each syndrome s we denote with $w(s)$ the Hamming weight of the coset leader $e(s)$. Let assume that the pairs $(s, w(s))$ for the code $[n, k, d]$ are kept in a look-up table. The Lexicographic Construction is iterative algorithm that can be described in 3 steps. In the first step, using linear search over $(s, w(s))$, the algorithm finds the covering radius ρ. In the second step it picks arbitrary syndrome s with weight $w(s) = \rho$ and forms a new codeword with the construction $c_{k+1} = \left(1^{d-\rho} \left| 0^k \right| s\right)$. In the third step the algorithm builds the table $(s_{k+1}, w(s_{k+1}))$ from $(s, w(s))$.

Given two syndromes s_k and s. The *companion set* of the syndrome s_k with respect to the syndrome s is the set:

$$K_{s_k} = \left\{ y_i \mid y_i = s_k + i \cdot s, i \in F_q \right\}. \tag{7}$$

We use the concept of companion sets to easily explain the creation of the new table $(s_{k+1}, w(s_{k+1}))$:

Theorem 6[4]: Given ρ, s_k, and $(s, w(s))$. The table $(s_{k+1}, w(s_{k+1}))$ can be constructed with the following minimization:

$$w(s_{k+1}) = \min_{\substack{y_i \in K_{s_{k+1}} \\ i \in F_q}} \left(wt\left(v + i^{d-\rho}\right) + w(y_i) \right) \tag{8}$$

for each syndrome $s_{k+1} = (v|s)$.

Theorem 7[4]: The space complexity of the Lexicographic Construction is:

$$\begin{cases} \Theta\left(\log(n) q^{H(\delta)n}\right) & за\, \delta \to const \\ \Theta\left(q^{H(\delta)n}\right) & за\, \delta \to 0 \end{cases}. \tag{9}$$

While the time complexity is $O\left(nq^{H(\delta)n}\right)$.

E. Wozencraft's ensemble

Wozencraft's ensemble is not a code, but an ensemble of codes $\{C_\alpha\}$ with code rate $R = 1/2$ [2,8]. The idea is to find a family of t disjoint sets $C_1, \ldots, C_t$ that partition the entire space F_q^n, such that each C_α is a linear subspace. If $t \ge V(n,d)$, then there exist at least one set C_α that is a linear code with parameters $\left[n, \log\left(|C_\alpha|\right), d\right]$. In addition, if we assume that all sets C_α are of same size, i.e. $|C_\alpha| = |C_\beta|$, $\forall \alpha, \beta \le t$, then the code dimension is easily determined, namely $k = \log\left(2^n / m\right)$.

For code rates $R = 1/2$ we construct the Wozencraft's ensemble as follows: for each α from $GF\left(2^k\right)$ we define the set C_α to be

$$C_\alpha = \left\{(x, \alpha x) \mid x \in GF\left(2^k\right)\right\}. \tag{10}$$

We can think of the sets C_α as linear $[n,k,?]$ codes with generator matrix $G = [I \quad \alpha I]$. Since there are 2^k disjoint sets C_α that cover the entire space F_2^n, the collection $\{C_\alpha\}$ is indeed Wozencraft's ensemble. In this ensemble there is at least one code with minimum distance d, such that d is the largest integer solution of

$$V(n,d) \le 2^{\frac{n}{2}}. \tag{11}$$

In order to find the best code in the Wozencraft's ensemble, for each $\alpha \in F_{2^{n/2}}$, first we construct the generator matrix $G = [I \quad \alpha I]$, then we find the minimum distance of the code. The time complexity of this approach is $O\left(n^c q^n\right)$.

III. Reed-Solomon Codes

Let each string m_i of m bits is interpreted as field element, namely $m_i \in GF\left(2^m\right)$. Then every message $M = [m_0 m_1 ... m_{K-1}]$ can be interpreted as polynomial

$$M(x) = m_0 + m_1 x + \ldots + m_{K-1} x^{K-1}. \tag{12}$$

Let α is the primitive element, then a codeword of the Reed-Solomon code is the N-touple

$$n = [n_0, n_2, \ldots, n_{N-1}],\ n \in GF\left(2^m\right) \tag{13}$$

where

$$n_i = M\left(\alpha^i\right). \tag{14}$$

Reed-Solomon codes are linear codes with minimum distance equal to $N - K + 1$. For each field $GF\left(q^m\right)$ and for each two N and K, such that $K \le N < q^m$, there exist a $[N, K, N-K+1]$ Reed-Solomon code.

IV. Concatenated Codes

Even though, in binary case, greedy algorithms produce codes with best-known parameters, they are considered impractical due to their exponential complexity with respect to the code length. Moreover no special-case poly-time algorithm that meets (6) has been designed to date nor has its non-existence been proved. Faced with this difficulty we are willing to accept codes with parameters that lag behind (6). Thus we say that the code $[n, Rn, \delta n]$ is *asymptotically good* if $R\delta > 0$.

Concatenation is code construction technique that produces sub-optimal, but asymptotically good codes. The simplest example of concatenation is the conversion of the $[N, K, N-K+1]_{2^m}$ Reed-Solomon code into binary code by encoding each field element of $GF\left(2^m\right)$ with a good $[n,m,d]_2$ binary code. The resulting concatenated code has parameters $\left[nN, mK, d(N-K+1)\right]_2$. If a greedy search is used to produce the code $[n,m,d]_2$, then following the terminology from [2] we call these codes *Forney's codes*.

Let $R_{RS} = K/N$ and $\delta_{RS} = D/N$ are the code rate and the relative distance of the RS code. Let $R_{GV} = k/n$ and $\delta_{GV} = d/n$ are the code rate and the relative distance of the greedy code. Then the concatenated code has relative distance

$$\delta = \delta_{GV} \cdot \delta_{RS} \tag{15}$$

and code rate

$$R(\delta, \delta_{GV}) = \left(1 - H(\delta_{GV})\right)\left(1 - \frac{\delta}{\delta_{GV}}\right). \tag{16}$$

Given δ, the best code rate $R(\delta)$ is obtained with the maximization

$$R(\delta) = \max_{\delta \le \delta_{GV} \le 1/2} \{R(\delta, \delta_{GV})\}. \quad (17)$$

$R(\delta)$ is known as the *Zyablov's bound*. Figure 1 shows the Zyablov bound compared with the Gilbert-Varshamov bound, and the gap between the concatenated codes and the greedy codes.

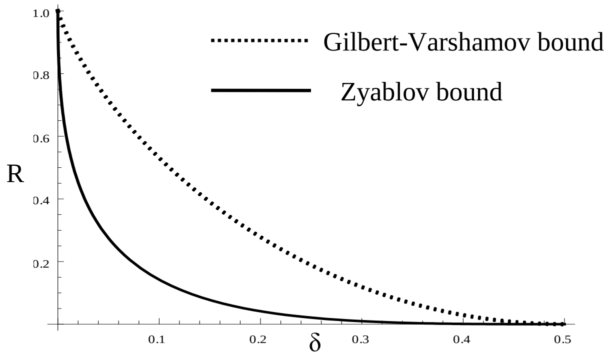

Figure 1: Zyablov bound.

V. Conclusion

If we use $[N, K, N-K+1]_{2^m}$ Reed-Solomon code, s.t. $N = 2^m - 1$, as the outer code and greedily search for the inner code then the resulting code construction will be polynomial with respect to N. In [2] M. Sudan suggested that the best code in Wozencraft's ensemble to be found and used as an inner code, thus lowering the complexity of the brute-force search for the inner code. From Section II we can observe that the complexity of finding the best Wozencraft's code is the same as the complexity of the Jenkins' construction. However, Jenkins' construction has two advantages: 1) finite-field operations are avoided and 2) the obtained code rates is not restricted to 1/2. Moreover, from section II, we can see that we can further lower the complexity of producing the inner code by using either the Lexicographic construction or the Varshamov algorithm with exponential space complexity. However, even with the use of these new greedy algorithms, the overall code construction is not totally explicit. Totally explicit constructions can be achieved by using all 2^m codes from the Wozencraft's ensemble. This construction is known as *Justesen codes* [9].

References

[1] G. D. Forney, "Concatenated Codes," PhD Thesis, Massachusetts Institute of Technology, 1965

[2] M. Sudan. (2008) http://courses.csail.mit.edu/6.440/spring08/index.html.

[3] Л. А. Бассалыго, В. В. Зяблов, and М. С. Пинскер, "ПРОБЛЕМЫ СЛОЖНОСТИ В ТЕОРИИ КОРРЕКТИРУЮЩИХ КОДОВ," *ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ*, vol. 13, no. 3, pp. 5-17, 1977

[4] D. Spasov, *Some Notes on the Good Codes*, Ss. Cyril and Methodius University, 2010

[5] B. Jenkins. (2002) Tables of Lexicodes. [Online]. "http://burtleburtle.net/bob/math/lexicode.html"

[6] A. Trachtenberg, "Designing Lexicographic Codes With a Given Trellis Complexity," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 89

[7] D. Spasov, *Implementing the Lexicographic Construction*, MS project, Boston University, 2006. Available at: http://nislab.bu.edu/nislab/projects/lexicode/index.html

[8] J. L. Massey, "Threshold Decoding," MIT, Cambridge, MA, Technical Report 410, 1963.

[9] J. Justesen, "Class of constructive asymptotically good algebraic codes." *IEEE Transactions on Information Theory*, vol. 18, no. 5, pp. 652–656, Sep 1972.