NONLINEAR q-ARY CODES: CONSTRUCTIONS AND MINIMUM DISTANCE COMPUTATION

MERCÈ VILLANUEVA, FANXUAN ZENG, AND JAUME PUJOL

ABSTRACT. A nonlinear code can be represented as the union of cosets of a linear subcode. Properties and constructions of new codes from given ones in terms of this representation can be described. Algorithms to compute the minimum distance of nonlinear codes, based on known algorithms for linear codes, are also established. Moreover, the performance of these algorithms is studied and an estimation of the number of enumerated codewords needed in the computations is given.

INTRODUCTION

Let $\mathbb{F}_q$ be a finite field with q elements and let $\mathbb{F}_q^n$ be the set of all vectors of length n over $\mathbb{F}_q$. The *Hamming distance* $d(u,v)$ between $u,v \in \mathbb{F}_q^n$ is the number of coordinates in which u and v differ. The *Hamming weight* $wt(u)$ of $u \in \mathbb{F}_q^n$ is $wt(u) = d(u,\mathbf{0})$, where $\mathbf{0}$ is the all-zero vector of length n. An (n,M,d) *q-ary code* C is a subset of $\mathbb{F}_q^n$ with M codewords and minimum distance d. The vectors of a code are called *codewords* and the *minimum distance*, denoted by $d(C)$, is the minimum value of $d(u,v)$ for all $u,v \in C$ and $u \neq v$.

Two q-ary codes C_1 and C_2 of length n are said to be *equivalent* if there is a vector $a \in \mathbb{F}_q^n$, a monomial matrix $\mathcal{M}$ and an automorphism Γ of the field $\mathbb{F}_q$ such that $C_2 = \{\mathcal{M}\Gamma(c)+a \,:\, c \in C_1\}$. Note that two equivalent codes have the same minimum distance. If C is linear, then $\mathbf{0} \in C$; but if C is nonlinear, then $\mathbf{0}$ does not need to belong to C. In this case, we can always consider a new code $C' = C - c$ for any $c \in C$, which is equivalent to C, such that $\mathbf{0} \in C'$. Therefore, from now on, we assume that $\mathbf{0} \in C$.

Given a q-ary code C, the problem of storing C in memory is a well known problem. If C is linear, that is, it is a subgroup of $\mathbb{F}_q^n$, then it can be compactly represented using a generator matrix. On the other hand, if C is nonlinear, it can be seen as the union of cosets of a linear subcode of C [7]. This allows us to represent a code as a set of representative codewords instead of as a set with all codewords.

Computing the minimum distance of a q-ary code C is necessary in order to establish its error-correcting capability. However, this problem is computationally difficult, and has been proven to be NP-hard [4]. If C is linear, $d(C)$ coincides with the minimum weight, denoted by $wt(C)$, and the Brouwer-Zimmermann minimum weight algorithm for linear codes over finite fields [1, 6] can be used. This algorithm can be found implemented in the computational algebra system MAGMA [2, 3, 5]. On the other hand, if C is nonlinear, $wt(C)$ and $d(C)$ do not always coincide, and as far as we know there is not any algorithm to compute them comparable to Brouwer-Zimmermann algorithm for linear codes.

This work has been partially supported by the Spanish MICINN under Grants TIN2010-17358 and TIN2013-40524-P, and by the Catalan AGAUR under Grant 2009SGR1224.

1. Representation and Construction of Nonlinear Codes

The *kernel* of a q-ary code C is defined as $K_C = \{x \in \mathbb{F}_q^n : \lambda x + C = C \ \ \forall \lambda \in \mathbb{F}_q\}$ [7]. Since $\mathbf{0} \in C$, K_C is a linear subcode of C. We denote by κ the dimension of K_C. In general, C can be written as the union of cosets of K_C, and K_C is the largest such linear code for which this is true [7]. Therefore, $C = \bigcup_{i=0}^{t}(K_C + v_i)$, where $v_0 = \mathbf{0}$, $t+1 = M/q^{\kappa}$, $M = |C|$ and $L = \{v_1, \ldots, v_t\}$ is the set of coset leaders. Note that for binary codes, $t \neq 1$, because if $t = 1$, $C = K_C \cup (K_C + v_1)$, but then C would be linear, so $C = K_C$. It is also important to emphasize that the coset leaders in this paper are not necessarily the ones having minimum weight in each coset. Since K_C is linear, it can be compactly represented by its generator matrix G of size $\kappa \times n$. Then, since the kernel takes up a memory space of order $O(n\kappa)$, the kernel plus the t coset leaders take up a memory space of order $O(n(\kappa + t))$.

Using this representation, we can manipulate and construct new nonlinear codes from old ones in a more efficient way. Specifically, it is possible to show how to establish the equality and inclusion of two given nonlinear codes from their kernels and coset leaders, and how to compute the kernel and coset leaders of new codes (union, intersection, extended, punctured, shorten, direct sum, Plotkin sum) from given ones, which are already represented in this way. All these results can be written to be implemented easily as algorithms. We can obtain the kernel and coset leaders of an extended code directly from the kernel and coset leaders of the code. The same happens for the direct sum and Plotkin sum constructions. For all other constructions, we obtain a partial kernel and the corresponding coset leaders. Although we can not assure which are the final kernel and coset leaders in these cases, we can speed up the kernel computation by starting from a partial kernel.

2. Minimum Distance Computation

The best known enumerative algorithm for linear codes to compute the minimum weight is the Brouwer-Zimmermann algorithm, which is based on the next result [5]. Let G be a generator matrix of a linear code K of dimension κ over $\mathbb{F}_q$. Any set of κ coordinates such that the corresponding columns of G are linear independent is called an *information set* for K. Let $G_1, \ldots, G_h$ be h systematic generator matrices of K such that they have pairwise disjoint information sets. For any $r < \kappa$, if $S_i = \{mG_i : m \in \mathbb{F}_q^k,\ wt(m) \leq r\}$ for each matrix G_i, then all $c \in C \setminus \bigcup_{i=1}^{h} S_i$ satisfy $wt(c) \geq h(r+1)$. After the rth step, we obtain a lower bound $h(r+1)$ and an upper bound of the minimum weight, which is the minimum weight of the enumerated codewords. When the two bounds meet, we obtain $wt(K)$ without enumerating necessarily all codewords. An adaption of this algorithm enables the use of more generator matrices with overlapping information sets, which means that the lower bound can grow faster during the enumeration process.

Given a q-ary linear code K of dimension κ and a vector $v \in \mathbb{F}_q^n \setminus K$, the linear span $K_v = \langle K, v\rangle = \bigcup_{\lambda \in \mathbb{F}_q}(K + \lambda v)$ of dimension $\kappa + 1$ is called an *extend coset*.

Proposition 2.1. *Let* $C = \bigcup_{i=0}^{t}\left(K_C + v_i\right)$. *The minimum distance* $d(C)$ *can be computed as* $\min(\{wt(K_{v_j - v_i}) \ : \ i \in \{0, 1, \ldots, t-1\},\ j \in \{i+1, \ldots, t\}\})$, *where* $v_0 = \mathbf{0}$.

Using the representation of nonlinear codes given in Section 1, Proposition 2.1 and the Brouwer-Zimmermann algorithm, we can design a new algorithm (MinD) to compute the minimum distance of a nonlinear code C, based on computing $wt(K_{v_j - v_i})$ using the known Brouwer-Zimmermann algorithm. Note that the complexity of this algorithm depends strongly on the number of coset

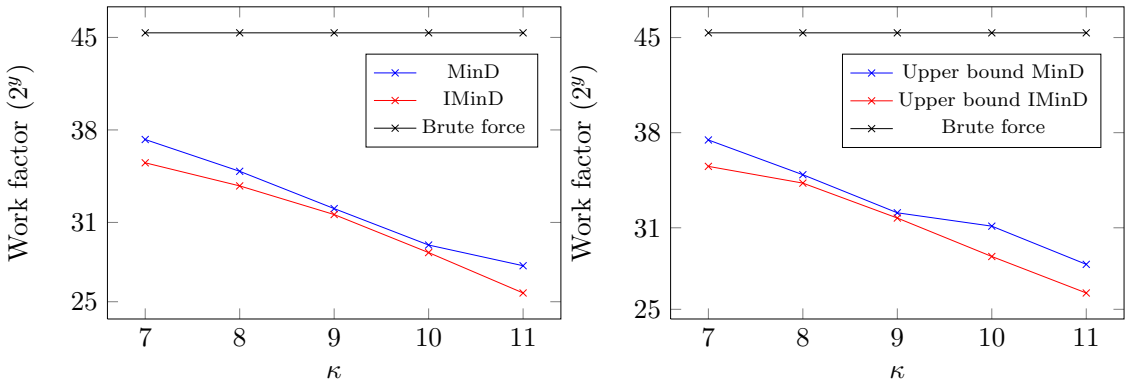

FIGURE 1. Work factors and upper bounds for computing $d(C)$ of ternary nonlinear codes C of length $n = 100$ and size $M = 3^{11} \cdot 6$.

leaders t and the complexity of the Brouwer-Zimmermann algorithm, since we need to compute $\binom{t+1}{2}$ times the minimum weight of a linear code.

This Algorithm MinD is based on the enumeration of codewords, adding together codewords and determining their minimum weight. The nature of these computations gives rise to a natural performance measure, which is referred to as *work* [5]. One unit of work represents both an addition and the weight computation of a single coordinate position. An estimate of the total work an algorithm performs is referred to as *work factor*. Therefore, work factors provide us with a tool for comparing the performance of algorithms based on enumeration. For example, note that the work factor for computing the minimum distance using a brute force algorithm is $\log_2(q)n\binom{q^\kappa(t+1)}{2}$.

Proposition 2.2. *Let C be a nonlinear code of length n with kernel of dimension κ and coset leaders $\{v_1, \ldots, v_t\}$. The work factor for computing $d(C)$ using Algorithm MinD is*

$$\sum_{i=0}^{t-1}\Bigg(\sum_{j=i+1}^{t}\Bigg(\log_2(q)(n-\kappa-1)\lceil n/(\kappa+1)\rceil\sum_{r=1}^{\bar{r}_{i,j}}\binom{\kappa+1}{r}(q-1)^{r-1}\Bigg)\Bigg) \tag{1}$$

where $\bar{r}_{i,j}$ is the smallest integer such that $\lfloor n/(\kappa+1)\rfloor(\bar{r}_{i,j}+1)+\max(0,\bar{r}_{i,j}+1-(\kappa+1-n \bmod (\kappa+1))) \geq wt(K_{v_j-v_i})$.

Note that the work factor for computing $d(C)$ relies on the parameters $\bar{r}_{i,j}$, which depend on $wt(K_{v_j-v_i})$, and they may be different for any i, j. Therefore, it is impossible to estimate the work factor if only the values n, κ and t are given. However, we can consider an upper bound of the work factor, and from that be able to estimate easily the work factor for computing $d(C)$. Since for any extend coset K_v we have that $wt(K_v) \leq wt(K_C)$, we can obtain an upper bound by replacing $wt(K_v)$ with $wt(K_C)$.

Proposition 2.3. *Let C be a nonlinear code of length n with kernel K_C of dimension κ and t coset leaders. An upper bound for the work factor of computing $d(C)$ using Algorithm MinD is given by*

$$\binom{t+1}{2}\log_2(q)(n-\kappa-1)\lceil n/(\kappa+1)\rceil\sum_{r=1}^{\bar{r}}\binom{\kappa+1}{r}(q-1)^{r-1} \tag{2}$$

where $\bar{r}$ is the smallest integer such that $\lfloor n/(\kappa+1) \rfloor (\bar{r}+1) + \max(0, \bar{r}+1-(\kappa+1-n \mod (\kappa+1))) \geq wt(K_C)$.

From Algorithm MinD, it is easy to see that the weight of some codewords in the kernel K_C is computed several times, specifically, once for each $K_{v_j - v_i}$, where $i, j \in \{0, 1, \ldots, t\}$ and $i < j$. Moreover, we need to compute the weight of extra vectors which belong to $K_C + \lambda(v_j - v_i)$, where $\lambda \in \mathbb{F}_q \backslash \{0, 1\}$. However, we can make a little adjustment to the algorithm, in order to avoid this extra computation. In Brouwer-Zimmermann algorithm, the enumerating process is divided into several steps. In the rth step, it enumerates all linear combinations of r rows of the generator matrix of $K_{v_j - v_i}$ of dimension $\kappa + 1$, examines the minimum weight of each combination and compares it with the lower bound. We can simplify this and enumerate only the codewords in each coset $K_C + v_j - v_i$. Then, in the rth step, we enumerate all linear combinations of r rows of the generator matrix of K_C of dimension κ and compute the weight of each combination adding the vector $v_j - v_i$. After this adjustment, the work factor using the improved Algorithm MinD, which is referred as Algorithm IMinD, can be reduced. Moreover, as before, we can also establish an upper bound for the work factor by using the same argument as in Proposition 2.3. Using this upper bound, again it is possible to estimate the work factor for computing $d(C)$ from the parameters n, κ, t and $wt(K_C)$ of a q-ary nonlinear code C. Note that the results on these upper bounds for the work factors allow to establish from which parameters of the given code, it is better to use the new presented algorithms instead of the brute force method.

Figure 1 shows the work factors (and the work factors upper bounds) for computing $d(C)$ using Algorithms MinD, IMinD and brute force, respectively. Note that when κ is large, Algorithms MinD and IMinD save a lot of time. Moreover, we can see the improvement on Algorithm MinD. In both tables, the work factors are expressed in logarithmic scale.

REFERENCES

[1] A. Betten, H. Fripertinger, A. Kerber, A. Wassermann, and K.-H. Zimmermann, *Codierungstheorie: Konstruktionen und Anwendung linearer Codes*, Berlin: Springer, 1998.

[2] J. J. Cannon and W. Bosma (Eds.), *Handbook of* MAGMA *Functions*, Edition 2.13, 4350 pages, 2006.

[3] M. Grassl, "Searching for linear codes with large minimum distance," in: W. Bosma and J. Cannon (Eds.) *Discovering Mathematics with Magma*, Springer, 2006.

[4] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1757-1773, 1997.

[5] G. White, *Enumeration-based Algorithms in Coding Theory*, PhD Thesis, University of Sydney, 2006.

[6] K.-H. Zimmermann, "Integral Hecke Modules, Integral Generalized Reed-Muller Codes, and Linear Codes," Tech. Rep. 3-96, Technische Universität Hamburg-Harburg, 1996.

[7] K. T. Phelps, J. Rifà, M. Villanueva, "Kernels and p-kernels of p^r-ary 1-perfect codes," *Designs, Codes and Cryptography*, 37, 243-261, 2001.

Universitat Autònoma de Barcelona
E-mail address: merce.villanueva@uab.cat, fanxuan@deic.uab.cat, jaume.pujol@uab.cat