Contributions to Discrete Mathematics

Volume 1, Number 1, Pages 1–7
ISSN 1715-0868

CLASSIFICATION OF LINEAR CODES EXPLOITING AN INVARIANT

STEFANO MARCUGINI, ALFREDO MILANI, AND FERNANDA PAMBIANCO

ABSTRACT. We consider the problem of computing the equivalence classes of a set of linear codes. This problem arises when new codes are obtained extending codes of lower dimension. We propose a technique that, exploiting a simply computed invariant, allows us to reduce the computational complexity of the classification process. Using this technique the $[13,5,8]_7$, the $[14,5,9]_8$ and the $[15,4,11]_9$ codes have been classified. These classifications enabled us to solve the packing problem for NMDS codes for $q = 7,8,9$. The same technique can be applied to the problem of the classification of other structures.

1. INTRODUCTION

Let F_q^n be the $n-$dimensional vector space over the Galois field F_q. The Hamming distance between two vectors of F_q^n is defined as the number of coordinates in which they differ. A $q-$ary linear $[n,k,d]_q-$code is a $k-$dimensional linear subspace of F_q^n with minimum distance d. For linear codes the minimum distance is equal to the minimum weight i.e. the minimum number of coordinates different from zero of a non-zero codeword.

This paper deals with the problem of classifying sets of linear codes. This problem arises, for example, using computer-based extension processes that construct new codes of dimension d_1 starting from codes of dimension d_2, $d_2 < d_1$. For examples of papers using such technique see [2], [4], [7] and [9]. In particular in [7] and [9], we constructed new near maximum-distance separable (NMDS) codes adding new rows to the generating matrix of NMDS codes of lower dimension. The starting step has been the classification of NMDS codes of dimension three obtained by geometrical means [8]. For a description of the properties of the NMDS codes see [3] and [4].

When extending a code in this way, several equivalent copies of the same code are obtained. A classification step allows us to compute the set of nonequivalent codes, but, when the number of examples to classify is high,

Received by the editors April 6, 2005, and in revised form, Dec. 7, 2005.

2000 *Mathematics Subject Classification.* 94-04, 94B05, 94B65.

Key words and phrases. Linear Codes, Classification, NMDS Codes.

This research is supported by Italian MIUR and GNSAGA.

©2006

some strategy has to be adopted to reduce the computational complexity of this phase.

The most direct and simple algorithm that can be used for the classification of a set S of codes keeps a list L of nonequivalent codes. Initially L is empty. All the codes C of S are considered: if there exists a code in L equivalent to C, then C is neglected, otherwise C is included in L. At the end L contains the set of representatives of the equivalence classes of S. The computational complexity of this simple algorithm is $O(|S| \times |L|)$, therefore it is practical only when $|S|$ and $|L|$ are relatively small. In [2], the program described in [1] was used. It deals with the problem of computing equivalence between codes exploiting invariants and signatures. In [6] a set of invariants was introduced allowing the equivalence of three dimensional binary codes to be determined.

To reduce the computational complexity of the classification step, we propose a technique of preclassification based on the use of an invariant. The condition on the invariant is that it must be easier to compute than the equivalence between two codes. In our case we used the minimum weight of the code.

Using the invariant in an opportune way, the set S is partitioned into subsets S_i such that $C_1 \in S_i$ and $C_2 \in S_j$ are not equivalent if $i \neq j$. It is then sufficient to classify the codes in each S_i separately. If each S_i contains only one equivalence class, the computational complexity of the classification step is $O(|S|)$. In our practical applications, most S_i's contain only one or few equivalence classes. There is an adjunctive cost, the computation of the invariant for the codes of S and for several truncated codes. This cost is negligible with respect to the cost of the classification phase.

Our technique is of general interest. In fact not only can different invariants be applied, but other computational classification problems can be faced, as long as there is a way to construct substructures preserving the invariant property.

The preclassification technique is described in Section 2. Section 3 contains some experimental results concerning the classification of the $[13,5,8]_7$, of the $[14,5,9]_8$ and of the $[15,4,11]_9$ codes. Section 4 contains concluding remarks regarding the general applicability of our preclassification technique and a list of results obtained. In particular we present a table describing the NMDS codes of maximal length for $q \leq 13$. Starting from the codes classified in this paper and applying fast extensions, duality and shortening, we determine the maximal length of an NMDS code in all the open cases for $q = 7, 8, 9$, solving therefore the packing problem in these cases. Using extension we also determine the maximal length of an NMDS code of dimension 4 for $q = 11$.

2. PRECLASSIFICATION USING AN INVARIANT

Our aim is the classification of a set of codes S. We consider equivalence in monomial sense, i.e. two $[n,k,d]_q$ codes C_1 and C_2, with respective generating matrices G_1 and G_2, are equivalent if there exist an invertible $(k,k)-$matrix A, an $(n,n)-$permutation matrix P and a field automorphism φ such that $G_1 = \varphi(AG_2P)$.

To reduce the number of the expensive computations of the equivalence between two codes, we use a numeric invariant, the minimum weight of the code. The problem of computing code minimum distance is known to be NP-hard (see e.g.[5]); however, for codes of small length and dimension (such as those considered here) the computation is easy.

We divide S into subsets S_i such that each code in S_i is of invariant value i. As mentioned above, it is desirable for each S_i to contain only one or perhaps a few equivalence classes. If the invariant is simple this will not be the case; however, we may use the same invariant to further subdivide each S_i. To do this, we exploit the fact that if C_1 and C_2 are equivalent $[n,k]$ codes and $\overline{C_1}$ is an $[n-1,k]$ code obtained by truncating C_1, then there exists a $[n-1,k]$ code $\overline{C_2}$ equivalent to $\overline{C_1}$ obtained by truncating C_2. This fact follows immediately from the definition of equivalence.

For each code C in each S_i we compute a first level index defined as the sum of the minimum weights of the n $[n-1,k]$ subcodes obtained truncating C by deleting a column of the generating matrix in all possible ways. If two codes C_1 and C_2 have different first level index, then they are not equivalent. In this way each S_i can be divided in subsets S_{i_j} such that $C_1 \in S_{i_j}$ and $C_2 \in S_{i_k}$ are not equivalent if $j \neq k$.

The process can be iterated, computing the second level index defined as the sum of the minimum weights of the $n*(n-1)/2$ $[n-2,k]$ subcodes obtained by truncating an $[n,k]$ code C deleting two columns of the generating matrix in all possible ways, and so on. Exploiting the indices of different levels, S is partitioned into subsets containing an ever-decreasing number of equivalence classes.

The computational cost of computing the index of order i of an $[n,k]$ code is $O\left(\binom{n}{i}\right)$. In the practical application we verify that it is sufficient to consider relatively small values of i to obtain sets of codes containing one or just a few numbers of equivalence classes. We note that two codes can have the same index of level i, but different indices of level j, $j < i$. Therefore when doing the preclassification it is useful to consider all indices belonging to the interval $[1,i]$ and not only the index of maximum value i.

3. EXPERIMENTAL RESULTS

This section describes the application of our preclassification technique for the classification of the $[13,5,8]_7$, $[14,5,9]_8$ and $[15,4,11]_9$ codes.

All computations have been done using MAGMA, a computer algebra package developed at the University of Sydney. The MAGMA function that verifies if two codes are equivalent is expensive, and the computational cost increases with the dimension of the codes. As our invariant we used the minimum weight of a code. In [7], extending the 923 nonequivalent $[11,3,8]_7$ codes, we obtained 80326 examples of $[13,5,8]_7$ codes such that any other $[13,5,8]_7$ code is equivalent to one of our examples. In an analogous way we obtained 4331 examples of $[14,5,9]_8$ codes and 69471 examples of $[15,4,11]_9$ codes extending respectively the 4181 $[12,3,9]_8$ codes and the 105193 $[14,3,11]_9$ codes found in [8].

Table 1 contains, for each set S of codes, the number of examples to classify, the number of classes obtained, the number of levels used in the preclassification step, the running time T_P, in hours, of the preclassification step, the running time T_C, in hours, of the classification step, and the ratio between the two running times. The duration of the preclassification does not exceed the duration of the classification step. The computation was done using a Sun Enterprise with a 450 MHz CPU.

Code	$\lvert S \rvert$	Classes	Levels	T_P	T_C	Ratio
$[13,5,8]_7$	80326	988	6	111	600	18.5%
$[14,5,9]_8$	4331	58	4	3.5	48	7.3%
$[15,4,11]_9$	69471	6585	5	140	168	83.3%

Table 1: Running time of the classification of the codes

Table 2 contains, for each set S of codes, the number of sets obtained in each level of the preclassification step. In the first and in the second case the preclassification was stopped when the number of sets obtained at the current level is almost equal to the number of codes of the previous level. Consequently, as seen in Table 3, most sets contained only one class. The computational cost associated with testing code equivalence is dependent on code dimension. The code in row 3 is of dimension 4, as such it was deemed inefficient to carry on the preclassification stage for this code past level 4. Cardinality is another index that can suggest whether or not a set contains many classes. At deeper levels only the sets whose cardinality exceeds a certain threshold could be further expanded. The threshold could be estimated considering the cardinalities of the smaller sets at the previous level. In this first implementation of this algorithm we did not use this feature.

		Level					
		0	1	2	3	4	5
Code	$[13,5,8]_7$	13	156	343	565	664	690
	$[14,5,9]_8$	13	39	49	55		
	$[15,4,11]_9$	16	196	681	1464	2570	

Table 2: Number of sets obtained at level k

Table 3 contains, for each set S of codes, the number of sets, obtained in the last step of the preclassification, containing k classes. In the first and second cases most sets contained one class. In the third case most sets contained a small number of classes. Hence, in all cases the computational cost of the classification step is near $O(|S|)$.

		Classes					
		1	2	3	$4-10$	$11-20$	$21-89$
Code	$[13,5,8]_7$	571	67	17	30	5	
	$[14,5,9]_8$	52	3				
	$[15,4,11]_9$	1690	365	160	262	56	37

Table 3: Number of sets of maximum level containing k classes

Table 4 shows the classification time expressed in hours, for the case of the fifty-two $[14,5,9]_8$ codes varying the number of levels of preclassification. The classification time in column 1 (where no preclassification is performed) is 58 times that in column 3. This corresponds exactly to the theoretic prevision. This computation has been done using a Pentium IV with a 2 GHz CPU.

	Levels				
	no preclassification	0	1	2	3
Time	831.27	109	28.36	19.26	14.35

Table 4: Classification of the $[14,5,9]_8$ codes using different levels of preclassification

4. Conclusions

We have proposed a technique for the classification of linear codes exploiting an invariant. As invariant we used the minimum weight of the code, but any invariant could be used in the same way. This technique could be also used for the classification of other structures, as far as there is a way to construct substructures preserving the invariant property. In this sense this is a general technique.

The classifications performed in this paper allowed us to determine the maximal length of an NMDS code for q=7,8,9 in all the remaining open cases, using extensions, duality and shortening. Starting from the classification of the 15 non-equivalent $[20,3,17]_{11}$ NMDS codes, we also demonstrated using an extension process that no $[21,4,17]_{11}$ code exists. Therefore the maximal length of an NMDS code of dimension 4 for $q=11$ is 20.

The following table contains what is currently known regarding the function $m'(k,q)$, representing the maximum length n for which there exists an $[n,k]_q$ NMDS code, for small values of k and q. The superscripts indicate the number of nonequivalent NMDS codes with the given parameters. The codes obtained in this paper and some other codes will be described in a forthcoming paper. See[9],[10] and [11] for the other references.

	q								
k	2	3	4	5	7	8	9	11	13
2	6^1	8^1	10^1	12^1	16^1	18^1	20^1	24^1	28^1
3	7^1	9^1	9^3	11^2	15^1	15^{19}	17^4	21^2	23^7
4	8^1	10^1	10^2	12^1	14^3	16^2	16^{19}	20	$21-24$
5		11^1	11^1	11^{60}	13^{988}	15^3	16^1	$18-21$	$21-25$
6		12^1	12^1	12^{31}	13	14	16	$18-22$	$21-26$
7			9^1	11^6	14	15	17	$18-23$	$21-27$
8			10^1	12^1	13^{988}	16	18	$18-24$	$21-28$
9				11^1	13^{294}	14^{58}	19	$19-25$	$21-29$
10				12^1	14^3	15^3	20	$20-26$	$21-30$
11					14^4	15^4	16^1	$18-27$	$21-31$
12					15^1	16^2	16^{19}	$18-28$	$21-32$
13					15^1	15^2	16^{382}	$18-29$	$21-33$
14					16^1	16^2	17^4	$18-30$	$21-34$
15						17^1	17^2	$18-31$	$21-35$
16						18^1	18^2	$20-32$	$21-36$

Bounds on $m'(k,q)$

Acknowledgment

The authors would like to thank Prof. D. Betten for his hints and suggestions.

References

[1] I. Bouyukliev, Q-extension – strategy in algorithms, in *Proc. of the 7th International Workshop on Algebraic and Combinatorial Coding Theory*, Bansko, Bulgaria (2000), 84-88.

[2] I. Bouyukliev and J. Simonis, Some new results on optimal codes over F_5, *Des. Codes Cryptography* **30** (2003), 97-111.

[3] S.M. Dodunekov and I. Landjev, On near-MDS codes, *J. Geometry* **54** (1995), 30-43.

[4] S.M. Dodunekov and I. Landjev, Near-MDS codes over some small fields, *Discrete Math.* **213** (2000), 55-65.

[5] I. Dumer, D. Micciancio and M. Sudan, Hardness of approximating the minimum distance of a linear code, *IEEE Trans. Inform. Theory* **49**(1) (2003), 22-37.

[6] J. Maks and J. Simonis, Polynomial invariants for binary linear codes, in *Proc. WCC 2001, International Workshop on Coding and Cryptography*, Paris, France (2001), 365-372.

[7] S. Marcugini, A. Milani and F. Pambianco, Existence and classification of NMDS codes over $GF(5)$ and $GF(7)$, in *Proc. of the 7th International Workshop on Algebraic and Combinatorial Coding Theory*, Bansko, Bulgaria (2000), 232–239.

[8] S. Marcugini, A. Milani and F. Pambianco, Classification of the [n,3,n-3]q NMDS codes over GF(7), GF(8) and GF(9), *Ars Combinatoria* **61** (2001) 263-269.

[9] S. Marcugini, A. Milani and F. Pambianco, NMDS Codes of Maximal Length over F_q, $8 \leq q \leq 11$, *IEEE Trans. Inform. Theory* **48**(4) (2002), 963-966.

[10] S. Marcugini, and F. Pambianco, AMDS codes of small dimension, in *Proc. of the 9th International Workshop on Algebraic and Combinatorial Coding Theory*, Kranevo, Bulgaria (2004), 277-282.

[11] J. Olsson, Linear Codes with Performance close to the Singleton Bound, Ph.D. Dissertation No. 605, Dept. Elec. Eng., Linkőping Univ., Sweden, 1999.

DIPARTIMENTO DI MATEMATICA E INFORMATICA, UNIVERSITÀ DEGLI STUDI DI PERUGIA, VIA VANVITELLI 1, 06123 PERUGIA ITALY
E-mail address: gino@dipmat.unipg.it

DIPARTIMENTO DI MATEMATICA E INFORMATICA, UNIVERSITÀ DEGLI STUDI DI PERUGIA, VIA VANVITELLI 1, 06123 PERUGIA ITALY
E-mail address: milani@dipmat.unipg.it

DIPARTIMENTO DI MATEMATICA E INFORMATICA, UNIVERSITÀ DEGLI STUDI DI PERUGIA, VIA VANVITELLI 1, 06123 PERUGIA ITALY
E-mail address: fernanda@dipmat.unipg.it