Implementation of an algorithm to compute the strong apparent distance of bivariate codes

D H Bueno-Carreño¹ and J A Medina López¹

¹ Pontificia Universidad Javeriana, Cali, Colombia

E-mail: dhbueno@javerianacali.edu.co

Abstract. The BCH bound is the oldest lower bound for the minimum distance of a cyclic code. The study of this bound and its generalizations are classical topics, which includes the study of the very well-known family of BCH codes. In 1970, P. Camion extended the notion of BCH bound to the family of abelian codes by introducing the apparent distance of polynomials. Camion showed that the minimum value of the apparent distance of certain polynomials associated to codewords is less than or equal to the minimum distance of the code. The mentioned minimum value is known as the apparent distance of an abelian code. In 2016, Bernal-Bueno-Simón introduced the notion of strong apparent distance of polynomials and hypermatrices and developed an algorithm to compute the minimum strong apparent distance of a hypermatrix based on $q$-orbits manipulations. In this work, we will present the implementation of an algorithm to compute the strong apparent distance of bivariate codes.

1. Introduction

Information theory was introduced by C. E. Shannon in the paper entitled “A Mathematical Theory of Communication” (1948) and studies, from a mathematical point of view, the problem of the transmission of a message from a transmitter to a receiver through a communication channel [1]. Error correcting coding theory (Coding theory, for short) is one of the most important branches of Information theory, in which this work is located. Coding theory develops methods to protect information against a noise, i.e., in a channel with noise the message sent is encoded in order that the receiver can correct, or at least detect, the possible errors that appear in the received message with respect to the original one [2–5].

The error-correcting capability of a code is directly related to its minimum distance and the problem of computing the minimum distance of any code is still open. In order to approach the solution to this problem some bounds have been proposed, particularly for the case of algebraic codes. The BCH bound is the oldest lower bound for the minimum distance of a cyclic code (see [6]). In 1970, P. Camion [7] extended the notion of BCH bound to the family of abelian codes by introducing the apparent distance of polynomials. Camion showed that the minimum value of the apparent distance of certain polynomials associated to codewords is less than or equal to the minimum distance of the code. The mentioned minimum value is known as the apparent distance of a code. In [8], Bernal-Bueno-Simón strengthened the notion of apparent distance, by introducing the notion of strong apparent distance and presented an algorithm to compute the strong apparent distance of an abelian code, based on some manipulations of hypermatrices associated to its generating idempotent. The aforementioned method computes...
this bound with fewer computations than those given in [7] and [9]. Furthermore, in the bivariate
case, the order of the computations is reduced from exponential to linear.

In this work, we present an implementation of [8, Algorithm 1] in SageMath [10], the results
of an evaluation of the typical computational costs of our approach and the measurement of the
number of iterations that Algorithm 1 needs to compute the minimum strong apparent distance
of a matrix. It is worth mentioning that this work is an advance of our research about the
relationship between Sidon sets and \( q \)-orbits.

2. Notation and preliminaries
This section introduces notation and terminology needed to understand our results. Also, it
recalls some basic definitions and facts. Throughout this paper, \( \mathbb{F}_q \) denotes the field with \( q \)
elements where \( q \) is a power of a prime \( p \). A bivariate abelian code is an ideal in the algebra
\( A_q(r_1, r_2) = \mathbb{F}_q[x, y]/(x^{r_1} - 1, y^{r_2} - 1) \).

We denote by \( I \) the set \( \mathbb{Z}_{r_1} \times \mathbb{Z}_{r_2} \) and we write the elements \( f \in A_q(r_1, r_2) \) as \( f = f(x, y) = \sum a_i X^i \), where \( i = (i_1, i_2) \in I \) and \( X^i = x^{i_1} y^{i_2} \). We deal with abelian codes in
the semisimple case; that is, we always assume that \( \gcd(r_1, q) = 1 \) and \( \gcd(r_2, q) = 1 \).

We denote by \( U_{r_i} \) the set of \( r_i \)-th primitive roots of unity and we define the set \( U \) by Equation 1

\[
U = \{ \alpha = (\alpha_1, \alpha_2) : \alpha_1 \in U_{r_1} \land \alpha_2 \in U_{r_2} \} \tag{1}
\]

Let \( \mathbb{F}_{q^\ell} \mid \mathbb{F}_q \) be an extension field containing \( U_{r_1} \) and \( U_{r_2} \), it is well known that every abelian
code \( C \) in \( A_q(r_1, r_2) \) is totally determined by its set of zeros given by Equation 2

\[
Z(C) = \{ \alpha \in \mathbb{F}_{q^\ell}^2 : f(\alpha) = 0 \text{ for all } f \in C \text{ and } \alpha^{(r_1, r_2)} = 1 \} \tag{2}
\]

For fixed \( \alpha = (\alpha_1, \alpha_2) \in U \), the code \( C \) is determined by its defining set, with respect to \( \alpha \),
which is defined as in Equation 3

\[
D_\alpha(C) = \{ (a_1, a_2) \in I : f(\alpha_1^{a_1}, \alpha_2^{a_2}) = 0 \text{ for all } f \in C \} \tag{3}
\]

Given an element \( a = (a_1, a_2) \in I \), we define its \( q^t \)-orbit modulo \( (r_1, r_2) \) as in Equation 4

\[
Q_t(a) = \left\{ (a_1 \cdot q^{t_i}, a_2 \cdot q^{t_i}) \in I : i \in \mathbb{N} \right\} \tag{4}
\]

In the case \( t = 1 \) we only write \( Q(a) \).

In the semisimple case, it is easy to see that for every abelian code \( C \) in \( A_q(r_1, r_2) \), the
defining set \( D_\alpha(C) \) is a disjoint union of \( q \)-orbits modulo \( (r_1, r_2) \) for a fixed \( \alpha \in U \). Conversely,
every union of \( q \)-orbits modulo \( (r_1, r_2) \) determines an abelian code (an ideal) in \( A_q(r_1, r_2) \) (see,
for example, [11]).

For any \( i \in I \) we write its \( k \)-th coordinate as \( i(k) \). A matrix with entries in a set \( R \) indexed
by \( I \) is a 2-dimensional \( I \)-array, denoted by \( M = (a_i)_{i \in I} \), with \( a_i \in R \) [12]. The set of indices,
the dimension and the ground field will be omitted if they are clear by the context. In the case
\( s = 1, M \) is a vector. We write \( M = 0 \) when all its entries are 0 and we write \( M \neq 0 \) otherwise.
For matrix \( M \), we define its columns as:
\[ H_M(1, b) = \{ a_i \in M : i(1) = b \} \]
\[ H_M(2, b) = \{ a_i \in M : i(2) = b \} \]  

(5)

with \( k \in \{1, 2\} \) and \( 0 \leq b < r_k \), where \( a_i \in M \) means that \( a_i \) is an entry of \( M \). A column will be seen as an 1-dimensional matrix (a vector). If \( M \) is a vector, we use the term entries. Note that in Equation 5, usually, \( H_M(1, b) \) is known as a row and \( H_M(1, b) \) is known as a column.

Let \( D \subseteq I \). The matrix afforded by \( D \) is defined as \( M = (a_i)_{i \in I} \) where \( a_i = 1 \) if \( i \notin D \) and \( a_i = 0 \) otherwise. When \( D \) is an union of \( q' \)-orbits we say that \( M \) is a \( q' \)-orbits matrix, and it will be denoted by \( M = M(D) \). For any \( I \)-matrix \( M \) with entries in a ring, we define the support of \( M \) as the set \( \text{supp}(M) = \{ i \in I : a_i \neq 0 \} \), whose complement will be denoted by \( D(M) \). Note that, if \( D \) is a union of \( q' \)-orbits then the \( q' \)-orbits matrix afforded by \( D \) verifies that \( D(M(D)) = D \). To define and compute the apparent distance of a bivariate code we will use the matrix afforded by its defining set, with respect to \( \alpha \in U \).

Let \( Q_t \) be the set of \( q' \)-orbits in \( I \), for some \( t \in \mathbb{N} \). We define a partial ordering over the set of \( q' \)-orbits matrices \( \{ M(D) : D = \cup Q_t \text{ for some } Q_t \in Q_t \} \) as in Equation 6.

\[ M(D) \leq M(D') \iff \text{supp}(M(D)) \subseteq \text{supp}(M(D')) \]  

(6)

Clearly, this condition is equivalent to \( D' \subseteq D \).

Let \( \mathbb{F}_{q'}|\mathbb{F}_q \) be an extension field such that \( U \subseteq \mathbb{F}_{q'}^2 \). The (discrete) Fourier transform of a polynomial \( f \in A_q(r_1, r_2) \), with respect to \( \alpha \in U \), is the polynomial \( \varphi_{\alpha,f}(X) = \sum_{j \in J} f(\alpha^j)X^j \in A_{q'}(r_1, r_2) \). It is known that the function Fourier transform may be viewed as an isomorphism of algebras \( \varphi_{\alpha} : A_q(r_1, r_2) \rightarrow (\mathbb{F}_{q'}^{|I|}, \ast) \), where the multiplication “\( \ast \)” in \( \mathbb{F}_{q'}^{|I|} \) is defined coordinatewise. So, we may see \( \varphi_{\alpha,f} \) as a vector in \( \mathbb{F}_{q'}^{|I|} \) or as a polynomial in \( A_{q'}(r_1, r_2) \).

3. The strong apparent distance

This section presents the notion of strong apparent distance of matrices. It also presents some technical results, some of them well-known, that are used to compute the strong minimum distance of abelian codes. For more details see [8].

3.1. The strong apparent distance

3.1.1. Definition. Let \( M \) be a matrix over \( \mathbb{F}_q \) and \( k \in \{1, 2\} \). For any \( k \in \{1, 2\} \) and \( b \in \mathbb{Z}_{r_k} \), the set of zero columns of \( M \) are the sets defined by Equation 7

\[ CH_M(1, b) = \{ H_M(k, b_0), \ldots, H_M(k, b_{k}) \} \quad \text{with } b = b_0, \]  

(7)

such that \( H_M(k, b_j) = 0 \) for all \( j \in \{0, \ldots, \ell\} \), \( b_0, \ldots, b_{\ell} \) is a list of consecutive integers modulo \( r_k \) and \( H_M(k, b_j) \neq 0 \). We denote by \( \omega_M(k, b) \) the value \( |CH_M(k, b)| \); in the case \( s = 1 \) we write \( \omega_M(b) = \omega_M(1, b) \).

Notice that for any \( k \in \{1, 2\} \) and \( b \in \mathbb{Z}_{r_k} \), we have that \( \omega_M(k, b) = 0 \) if and only if \( H_M(k, b) \neq 0 \).

3.1.2. Definition. Let \( q, r_1, r_2 \) and \( I \) be as above. Let \( M \) be a matrix over \( \mathbb{F}_q \) and \( k \in \{1, 2\} \). The strong apparent distance of \( M \), denoted by \( \text{sd}^*(M) \), is defined as follows:

1) \( \text{sd}^*(0) = 0 \).
2) if \( M \) is a vector, its strong apparent distance is \( \text{sd}^*(M) = \max\{\omega_M(b) + 1 : b \in \mathbb{Z}_r\} \).
3.1.3. Definition. Let $M$ be a nonzero matrix. We say that a pair $(k, b)$, where $k \in \{1, 2\}$ and $b \in \mathbb{Z}_{r_1}$, is an involved pair (in the computation of $sd^*(M)$) if $sd^*(M) = (\omega_M(k) + 1)sd^*(H_M(k, b))$. The column $H_M(k, b)$ is called, in turn, an involved column (in the computation of $sd^*(M)$). We denote the set of involved pairs by $Ip(M)$.

3.2. The strong apparent distance of an abelian code

Now, we are going to define the strong apparent distance of an abelian code in an analogous way to Camion’s definition [7] (see also [9, pp. 187-188]).

3.2.1. Definition. Let $C$ be a code in $A_q(r_1, r_2)$. The strong apparent distance of $C$, with respect to $\alpha \in U$, is $sd^*_\alpha(C) = \min \{sd^*(M(\varphi_{\alpha, e})) : 0 \neq e^2 = e \in C\}$, where $\varphi_{\alpha, e}$ denotes the image of $e$ under the discrete Fourier transform with respect to $\alpha$, as we denoted in Section 2, and $M(\varphi_{\alpha, e})$ denotes the coefficient matrix of $\varphi_{\alpha, e}$.

The strong apparent distance of $C$ is $sd^*(C) = \max \{sd^*_\beta(C) : \beta \in U\}$. We also define the set of optimized roots of $C$ as $R(C) = \{\beta \in U : sd^*(C) = sd^*_\beta(C)\}$.

Note that $\min \{sd^*(P) : 0 \neq P \leq M\} = \min \{sd^*(M(\varphi_{\alpha, e})) : 0 \neq e^2 = e \in C\} = sd^*_\alpha(C)$ [8]. This fact drives us to give the following definition.

3.2.2. Definition. In the setting described above, for a $q^t$-orbits hypermatrix $M$, its minimum strong apparent distance is $msd(M) = \min \{sd^*(P) : 0 \neq P \leq M\}$.

4. Computing the MSD of a hypermatrix

This section presents a technique to compute the minimum strong apparent distance of a matrix and thereby to compute the strong apparent distance of an abelian code. For more details see [8].

Let $q$, $r_1$, $r_2$ and $I$ be as in the preceding section, and let $Q_t$ be the set of $q^t$-orbits in $I$, for some $t \in \mathbb{N}$. For an arbitrary subset $Q' \subseteq Q_t$, we set $D = \bigcup_{Q \in Q'} Q$, and construct $M = M(D)$, the $q^t$-orbits matrix afforded by $D$. Consider an arbitrary column of $M$, say $H_M(k, b)$, where $k \in \{1, 2\}$ and $b \in \mathbb{Z}_{r_1}$. Then we define $D_M(k, b) = I(k, b) \setminus \text{supp}(H_M(k, b))$.

4.0.1. Proposition. ([8, Proposition 25]) Let $Q_t$ be the set of $q^t$-orbits modulo $(r_1, r_2)$, $\mu \in \{1, \ldots, |Q_t| - 1\}$ and $\{Q_j\}_{j=1}^{\mu}$ a subset of $Q_t$. Set $D = \bigcup_{j=1}^{\mu} Q_j$ and $M = M(D)$. Then there exist two sequences: the first one is formed by nonzero $q^t$-orbits matrices, $M = M_0 > \cdots > M_\ell \neq 0$ and the second one is formed by positive integers, $m_0 \geq \cdots \geq m_\ell$ with $\ell \leq \mu$ and $m_i \leq sd^*(M_i)$, for $0 \leq i \leq \ell$, verifying the following property: If $P$ is a $q^t$-orbits matrix such that $0 \neq P \leq M$, then $sd^*(P) \geq m_\ell$ and if $sd^*(P) < m_{i-1}$ then $P \leq M_i$, where $0 < i \leq \ell$. Moreover, if $\ell' \in \{0, \ldots, \ell\}$ is the first element satisfying that $m_{\ell'} = m_\ell$ then $sd^*(M_{\ell'}) = msd(M)$. The proof of the Proposition 4.0.1 provides a recursive construction that leads to the following algorithm.
4.0.2. Remark. Note that if the matrix has $\mu$ $q$-orbits, Algorithm 1 has at most $\mu$ steps.

Algorithm 1 (Matrices). Set $I = \mathbb{Z}_{r_1} \times \mathbb{Z}_{r_2}$. Consider the matrix $M = (m_{ij})_{(i,j) \in I}$.

**Step 1.** Compute the strong apparent distance of $M$ and set $m_0 = sd^*(M)$.

**Step 2.**

a) If there exists $H_M(k,b) \in Ip(M)$ such that $sd^*(H_M(k,b)) = 1$ then we finish giving the sequences $M = M_0$ and $m_0 = sd^*(M)$.

b) In the case $sd^*H_M(k,b) \neq 1$ for all $(k,b) \in Ip(M)$, we set

$$S = \bigcup_{(k,b) \in Ip(M)} supp(H_M(k,b))$$

and construct the matrix $M_1 = (a_{ij})_{(i,j) \in I}$ such that

$$a_{ij} = \begin{cases} 0 & \text{if } (i,j) \in \bigcup \{Q(k,b) : (k,b) \in S\} \\ m_{ij} & \text{otherwise.} \end{cases}$$

(we know that if $0 \neq P < M$ and $d^*P < m_0$ then $P \leq M_1$).

**Step 3.**

a) If $M_1 = 0$ then we finish giving the sequences $M = M_0$ and $m_0 = sd^*(M)$.

b) If $M_1 \neq 0$, we set $m_1 = \min\{m_0, sd^*(M_1)\}$, and we get the sequences $M = M_0 > M_1$ and $m_0 \geq m_1$. Then, we go back to Step 1 with $M_1$ instead of $M$ and $m_1$ instead of $m_0$.

5. Implementation

In order to test the performance of Algorithm 1, we implement it in SageMath [10]. The code is available on GitHub looking for minimum-strong-apparent-distance. Now, to evaluate the typical computational costs of our approach, we consider $q = 2$ and we generate pairs of values $(r_1, r_2)$ where $r_1 = 45$ and $r_2$ is a odd number such that $3 \leq r_2 \leq 101$. For each case, we construct the matrix $M(D)$ where $D$ is the union of five 2-orbits that are randomly selected. Then, we test the algorithm and measure the total CPU time used to get the minimum strong apparent distance. The algorithm runs on a Linux box with Intel Core i7-7500U at 2.7GHz. The results are the average of 1000 executions of Algorithm 1 and are summarized in Figure 1.

![Figure 1. CPU time used to get the minimum strong apparent distance.](image)
Now, in order to measure the number of iterations that Algorithm 1 needs to compute the msd for a hypermatrix $M$, we consider $r_1 = 101$, $r_2 = 105$ and $D = \bigcup_{i=1}^{\mu} Q_i$ where $2 \leq \mu \leq 40$. For each case, we construct the matrix $M = M(D)$ and we compute its msd. Figure 2 shows that the iteration used on each case are less than $\mu$ as Remark 1 said.

![Figure 2. Iterations used by Algorithm 1 to get the minimum strong apparent distance.](image-url)

6. Conclusions and future work
In general, the minimum strong apparent distance is not easy to compute by hand. So, computational tools are required to simplify the calculations with the aim to detect patterns at large scale. In this order of ideas, we found that the bound proposed to get the maximum number of iterations could be refined. As a future work, we are going to identify a possible relation between Sidon sets and $q$-orbits.

References

[1] Shannon C E 1948 *Bell system technical journal* **27** 379–423
[2] Berman S 1967 *Cybernetics* **3** 17–23
[3] Jensen J 1985 *IEEE transactions on information theory* **31** 788–793
[4] Pless V, Bruedl R A and Huffman W C 1998 *Handbook of coding theory* (Elsevier Science Inc.)
[5] Landrock P and Manz O 1992 *Designs, Codes and Cryptography* **2** 273–285
[6] Bose R C and Ray-Chaudhuri D K 1960 *Information and Control* **3** 68–79
[7] Camion P 1971 Abelian codes. Tech. rep. Wisconsin Univ. Madison Mathematics Research Center
[8] Bernal J J, Bueno-Carreño D H and Simón J J 2016 *IEEE Transactions on Information Theory* **62** 655–668
[9] Sabin R E 1992 *Applicable Algebra in Engineering, Communication and Computing* **3** 183–197
[10] The Sage Developers 2017 *SageMath, the Sage Mathematics Software System (Version 8.1)*
    http://www.sagemath.org
[11] Bernal J J and Simón J J 2011 *IEEE Transactions on Information Theory* **57** 7990–7999
[12] Yamada K 1965 *Hitotsubashi journal of arts and sciences* **6** 34–44