Calculating the Minimum Distance of a Toric Code via an Algebraic Algorithm

Fadime Baldemir[1][0000−0003−4941−1823] and Mesut Şahin[2][0000−0002−0310−2542]

[1] Çankırı Karatekin University, Çankırı, Turkey
fadimeozkan@karatekin.edu.tr
[2] Hacettepe University, Ankara, Turkey
mesut.sahin@hacettepe.edu.tr

Abstract. Toric codes are examples of evaluation codes. They are produced by evaluating polynomials of degree d at the points of a subset Y of a toric variety X. These codes reveal how algebraic geometry and coding theory are interrelated. The minimum distance of a code is the minimum number of nonzero entries in the codewords of the code.
Let I be the ideal generated by all homogeneous polynomials vanishing at all the points of Y, which is also known as the vanishing ideal of Y.
Our algorithm computes the minimum distance by finding a homogeneous polynomial f among all homogeneous polynomials of the same degree which has maximum number of roots on Y.

Keywords: Minimum distance · Toric code· Hilbert Function.

Extended Abstract

Let $\Sigma \subseteq \mathbb{R}^n$ be a complete simplicial fan with rays $\rho_1, \ldots, \rho_r$ and $X := X_\Sigma$ be the corresponding n dimensional smooth projective toric variety with Picard group isomorphic to $\mathbb{Z}^d$ over a finite field $\mathbb{F}_q$, where $d = r - n > 0$.

Toric varieties can be constructed starting from some nice combinatorial objects called fan. Every cone in Σ corresponds to an affine toric variety and these are glued to obtain the abstract normal toric variety X together along the affine toric variety they contain that corresponds to the intersections of the cones in the fan. Let $v_1, \ldots, v_r$ be the generators of the rays of the cones in the fan Σ and ϕ be the matrix whose rows are these generators of the rays. We get the matrix β in the following exact sequence by applying Smith Normal Form Algorithm to ϕ:

$$\mathfrak{B} : 0 \longrightarrow \mathbb{Z}^n \xrightarrow{\phi} \mathbb{Z}^r \xrightarrow{\beta} \mathcal{A} \longrightarrow 0$$

Here the group $\mathcal{A} \cong \mathbb{Z}^r/\phi(\mathbb{Z}^n)$, is isomorphic to the Picard group of X. Applying $Hom(-, \mathbb{K}^*)$ functor to the sequence $\mathfrak{B}$ gives the following short exact sequence:

$$\mathfrak{B}^* : 1 \longrightarrow G \xrightarrow{i} (\mathbb{K}^*)^r \xrightarrow{\pi} T_X \longrightarrow 1$$

with $\pi : (\xi_1, ..., \xi_r) \to (\xi^{u_1}, ..., \xi^{u_n})$ where $u_1, ..., u_n$ are the columns of ϕ.

Let $S = \mathbb{F}_q[x_1, ..., x_r] = \bigoplus_{\alpha \in \mathcal{A}} S_\alpha$ be the multigraded polynomial ring which is also known as the Cox ring of X graded using the columns of the matrix β, i.e. $deg_{\mathcal{A}}(x_j) = \beta_j$ is the j-th column of β, for $j = 1, ..., r$. The irrelevant ideal of S is the monomial ideal

$$B = \langle x^{\hat{\sigma}} : \quad \sigma \in \Sigma \rangle, \quad \text{where } x^{\hat{\sigma}} = \prod_{\rho_i \not\in \sigma} x_i.$$

The main feature of X we use is we can represent points on our toric variety X using homogeneous coordinates thanks to the Geometric Invariant Theory quotient representation

$$X \cong \mathbb{K}^r \setminus V(B))/G \text{ where } G = Ker(\pi) \text{ and } \mathbb{K} = \overline{\mathbb{F}}_q,$$

due to Cox, see [1]. Therefore, every point of X is identified with an orbit of the following type:

$$[P] = G \cdot P = [p_1 : \cdots : p_r], \text{ for } P \in \mathbb{K}^r \setminus V(B).$$

For a fixed degree α in the semigroup $\mathbb{N}\beta$ which is generated by $\beta_1, ..., \beta_r$, and a subset $Y = \{[P_1], ..., [P_N]\} \subseteq X$ with cardinality N, the evaluation map is defined as follows:

$$ev_Y : S_\alpha \mapsto \mathbb{F}_q^N, \, F \mapsto (F(P_1), \ldots, F(P_N))$$

The image of S_α is called a (generalized) toric code which is denoted by $\mathcal{C}_{\alpha,Y}$, see [4] for a survey. By definition, the size of Y is the *length* of the code and the *dimension* $\dim_{\mathbb{F}_q} \mathcal{C}_{\alpha,Y}$ of the code is the dimension of $\mathcal{C}_{\alpha,Y}$ as a vector space over $\mathbb{F}_q$. The number of nonzero entries in a codeword is called its *weight* and the *minimum distance* δ is obtained by calculating the smallest weight among all nonzero codewords.

It is clear that the kernel of the linear map ev_Y is the degree α part $I_\alpha(Y)$ of the β-graded or homogeneous *vanishing ideal* $I(Y)$ of Y, which is defined to be the ideal generated by homogeneous polynomials vanishing at all the points of Y. Due to this relation, the dimension $dim_{\mathbb{F}_q} \mathcal{C}_{\alpha,Y}$ equals the value $H_{I(Y)}(\alpha) := \dim S_\alpha - \dim I_\alpha(Y)$ of the multigraded *Hilbert function* $H_{I(Y)}$ of $I(Y)$. It is known that for sufficiently large values of α, the function $H_{I(Y)}$ agrees with a polynomial $P_{I(Y)}$ known as the multigraded *Hilbert polynomial* of $I(Y)$. If I is the B-saturated ideal corresponding to the set of ℓ points on a smooth projective toric variety, then Hilbert polynomial of I is just the constant $P_I(t) = \ell$ (see Example 4.12 in [5]). Therefore, the length of the code $\mathcal{C}_{\alpha,Y}$ is $P_{I(Y)}(t) = N$.

In the first step, we obtain the following result as in [2] in which case X is a projective space:

Lemma 1 *Let $Y \subseteq X$. Then, the minimum distance of the corresponding code is*

$$\delta_Y(\alpha) = N - max\{| \, V_{X,Y}(f) \, | : f \in S_\alpha \setminus I_\alpha(Y)\},$$

where $V_{X,Y}(f)$ is the subset of Y which consists of the roots of f.

By using the lemma above we can obtain the following main result.

Theorem 1. *Let $Y \subseteq X$. Then, the minimum distance of the corresponding code is*

$$\delta_Y(\alpha) = N - max\{P_{I(V_{X,Y}(f))} \mid : f \in S_\alpha \setminus I_\alpha(Y) \text{ is a zero-divisor}\},$$

where $P_{I(V_{X,Y}(f))}$ denotes the multigraded Hilbert Polynomial of the ideal $I(V_{X,Y}(f))$.

As a consequence, we obtain the following algorithm which calculates the minimum distance of a code obtained from a smooth projective toric variety:

Algorithm 1 Calculating the minimum distance of a toric code.

Input A prime power q, a toric variety X over the field $\mathbb{F}_q$, a degree α together with the vanishing ideal $I(Y)$ of Y.

Output The minimum distance $\delta_Y(\alpha)$.

1: Find a basis of the vector space $M_\alpha = S_\alpha/I_\alpha$ for $M = S/I(Y)$.
2: Form the set M_α by taking $\mathbb{F}_q$-linear combinations of the basis elements of M_α.
3: Determine zero-divisors $f \in M_\alpha$ by checking if $I(Y) : f \neq I(Y)$.
4: Find the primary decomposition of $I(Y)$.
5: Find the ideals $I(V_{X,Y}(f))$ for zero-divisors $f \in M_\alpha$.
6: Return $\delta_Y(\alpha) = P_{I(Y)} - max\{P_{I(V_{X,Y}(f)} : f \text{ is a zero-divisor }\}$.

It is worth mentioning that the vanishing ideal needed in the algorithm can be found as described in [3] when Y is a subgroup of the dense torus of X.

Procedure 1 *Calculating the minimum distance of a toric code with Macaulay2 [6].*

```
i4 : Balpha=basis(alpha,S/IY);

i5 : N=flatten applyTable({apply(toList (set(0..q-1))^**
       (hilbertFunction(alpha,S/IY))- (set{0})^**
       (hilbertFunction(alpha,S/IY)),i-> toList i)}, i-> deepSplice i);
     P= apply(#N, j-> vector flatten N_{j});
     D= for i from 0 to #P-1 list Balpha*  flatten P_{i};
     A= flatten for i from 0 to #D-1 list entries  (flatten D_{i})#0;
     Malpha= apply(A, i->substitute(i,S));

i6 : Z=select(Malpha, f-> not quotient(IY,ideal f)==IY);

i7 : PrIY=primaryDecomposition IY;

i8 : IVXYf=(PrIY,f,S) -> (int := ideal (1_S) ;
           scan(PrIY, i -> if  f%i==0 then int=intersect(int,i)) ; int);
     Ideals=apply(Z,f->IVXYf(PrIY,f,S));

i9 : delta = hilbertPolynomial(X,IY)
           - max apply(Ideals, I-> hilbertPolynomial (X,I))
```

Our algorithm makes use of the following Macaulay2 package adopted to our notation:

```
i1 : needsPackage "NormalToricVarieties";
i2 : ring NormalToricVariety := PolynomialRing =>
     (cacheValue symbol ring)
     ( X -> (
         if isDegenerate X then
             error "-- not yet implemented for degenerate varieties";
         if not isFreeModule classGroup X then
             error "-- gradings by torsion groups not yet implemented";
         -- constructing ring
         K := X.cache.CoefficientRing;
         x := X.cache.Variable;
         r := #rays X;
         deg := entries transpose fromWDivToCl X;
         S := K (monoid [x_1..x_r, Degrees => deg]);
         S.variety = X;
         S  ) );
```

Example 1 *Let X be the Hirzebruch Surface H_2 over $K = \mathbb{F}_7$. The first exact sequence above becomes: $\mathfrak{P} : 0 \longrightarrow \mathbb{Z}^2 \xrightarrow{\phi} \mathbb{Z}^4 \xrightarrow{\beta} \mathbb{Z}^2 \longrightarrow 0$, where*

$$\phi = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 2 & -1 \end{bmatrix}^T \quad \text{and} \quad \beta = \begin{bmatrix} 1 & -2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Thus, the Cox ring $S = K[x_1, x_2, x_3, x_4]$ of X is graded via

$$\deg_{\mathcal{A}}(x_1) = \deg_{\mathcal{A}}(x_3) = (1, 0), \deg_{\mathcal{A}}(x_2) = (-2, 1), \deg_{\mathcal{A}}(x_4) = (0, 1).$$

Consider the subset $Y \subset X$ with $I(Y) = (x_3^2 - x_1^2, x_4^3 - x_3^6 x_2^3)$.

Our inputs for the example are as follows:

```
i3 : q=7; alpha={0,1};
     X=hirzebruchSurface(2, CoefficientRing => GF(q,Variable=>t) );
     S=ring X;
     IY=ideal ((x_3)^(2)-(x_1)^(2),(x_4)^(3)-(x_3)^(2*(3))*x_2^(3));
```

Using Procedure 1, we compute the minimum distance of $C_{\alpha,Y}$ to be 3.

References

1. Cox, D. A., Little, J. B., Schenck, H. K.: Toric varieties (Vol. 124). American Mathematical Soc. (2011)
2. Martínez-Bernal, J., Pitones, Y., Villarreal, R. H.: Minimum distance functions of graded ideals and Reed–Muller-type codes. Journal of Pure and Applied Algebra **221**(2), 221–275 (2017)
3. Baran, E., Şahin, M.: On parameterized toric codes. Applicable Algebra in Engineering, Communication and Computing, 1–25 (2021)
4. Şahin, M.: Lattice Ideals, Semigroups and Toric Codes. Numerical Semigroups, 285–302 (2020)
5. Maclagan, D., Smith, G. G.: Uniform Bounds on Multigraded Regularity. Algebraic Geometry **14**, 137–164 (2005)
6. Grayson, D. R., Stillman, M. E. : Macaulay2, a Software System for Research in Algebraic Geometry (2002).