List decoding of a class of affine variety codes

Olav Geil, Casper Thomsen

List decoding of a class of affine variety codes

Olav Geil and Casper Thomsen

Department of Mathematical Sciences,Aalborg University, Denmark
olav@math.aau.dk, caspert@math.aau.dk

Abstract. Consider a polynomial F in m variables and a finite point ensemble $S = S_1 \times \cdots \times S_m$. When given the leading monomial of F with respect to a lexicographic ordering we derive improved information on the possible number of zeros of F of multiplicity at least r from S. We then use this information to design a list decoding algorithm for a large class of affine variety codes.

Keywords. Affine variety code, list decoding, multiplicity, Schwartz-Zippel bound.

1 Introduction

In this paper we study affine variety codes over $\mathbf{F}_q$ in the special case where the variety is $S_1 \times \cdots \times S_m$. Here, $S_i \subseteq \mathbf{F}_q$, $i = 1, \ldots, m$. More formally write $S = S_1 \times \cdots \times S_m = \{P_1, \ldots, P_{|S|}\}$ and consider the evaluation map

$$\mathrm{ev}_S : \mathbf{F}_q[X_1, \ldots, X_m] \rightarrow \mathbf{F}_q^{|S|}, \qquad \mathrm{ev}_S(F) = (F(P_1), \ldots, F(P_{|S|})).$$

Let $\mathbb{M} \subseteq \{X_1^{i_1} \cdots X_m^{i_m} \mid 0 \leq i_j < |S_j|, j = 1, \ldots, m\}$ and define the affine variety code

$$E(\mathbb{M}, S) = \mathrm{Span}_{\mathbf{F}_q}\{\mathrm{ev}_S(M) \mid M \in \mathbb{M}\}.$$

Clearly, the generalized Reed-Muller codes is a special example of such a code. Other well-known examples are the hyperbolic codes [6], Reed-Solomon product codes [1], (generalized) toric codes [10, 14, 3], and Joyner codes [11]. In the present work we show that the dimension of $E(\mathbb{M}, S)$ equals $|\mathbb{M}|$ and we present a lower bound on the minimum distance which turns out to be sharp when $|\mathbb{M}|$ satisfies certain reasonable criteria. We then present a decoding algorithm for $E(\mathbb{M}, S)$. The algorithm is a straightforward generalization of the Guruswami-Sudan decoding algorithm for Reed-Solomon codes [9], except that it involves a preparation step. A main ingredient in the preparation step is information about how many zeros of multiplicity at least r a polynomial $F(X_1, \ldots, X_m)$ can have given information about its leading monomial with respect to a lexicographic ordering.

It is well-known that the generalized Reed-Muller codes can be viewed as subfield subcodes of Reed-Solomon codes. As demonstrated by Pellikaan and Wu [12, 13] this observation leads to an efficient decoding algorithm of generalized Reed-Muller codes via the Guruswami-Sudan decoding algorithm for Reed-Solomon

codes. When the variety does not equal $S = \mathbf{F}_q \times \cdots \times \mathbf{F}_q$, or when $\mathbb{M}$ is not chosen as all the monomials of degree up to some number, such a neat result does not hold any more.
Pellikaan and Wu also presented a direct interpretation of the Guruswami-Sudan decoding algorithm for generalized Reed-Muller codes. Even though Augot et al. improved the analysis of the latter algorithm dramatically [1, 2]; the algorithm that uses the subfield subcode is still superior when decoding generalized Reed-Muller codes. The analysis by Augot et al. uses a generalization of the Schwartz-Zippel bound to also deal with multiplicity. A proof of this bound was given by Dvir et al. in [5]. Augot et al. then used their insight to generalize the second algorithm by Pellikaan and Wu (the direct interpretation of the Guruwami-Sudan algorithm) to Reed-Muller like codes over $S = S_1 \times \cdots \times S_m$, when $S_1 = \cdots = S_m$, and to Reed-Solomon product codes as well. Based on the generalized Schwartz-Zippel bound they estimated the decoding radius.
In the present work we improve upon the methods from [5] to derive more detailed information on how many zeros of prescribed multiplicity a polynomial can have. Rather than using only information on the degree of the polynomial we use information on the leading monomial with respect to a lexicographic ordering. We present both a recursive algorithm to find such bounds and also closed formula expressions for the case of polynomials in two variables.
The interpolation polynomial $Q(X_1, \ldots, X_m, Z)$ in the list decoding algorithm of the present paper can be viewed as a polynomial in Z with coefficients from $\mathbf{F}_q[X_1, \ldots, X_m]$. Having fixed the code to be used, in a preparation step we determine sets from which the coefficients must be taken. Here, we use our improved insight on how many zeros with prescribed multiplicity a polynomial can have given information about its leading monomial. The idea of a preparation step comes from [7] where an interpretation of the Guruswami-Sudan decoding algorithm without multiplicity was described for order domain codes. Our experiments show that the method of the present work is an improvement upon the situation where only the generalized Schwartz-Zippel bound is used for the design and the analysis of the decoding algorithm. Our algorithm works for codes of not too high dimensions. For small dimensions the algorithm often decodes more than $d/2$ errors.

2 Parameters of the codes

Throughout the paper we shall use the notation $s_i = |S_i|$ for $i = 1, \ldots, m$. Also we shall write $n = |S|$ as clearly the length of $E(\mathbb{M}, S)$ equals $|S|$. First we show how to find the dimension of the code.

Proposition 1. *The dimension of $E(\mathbb{M}, S)$ equals $|\mathbb{M}|$.*

Proof. We show that $\{\mathrm{ev}_S(X_1^{i_1}, \ldots, X_m^{i_m}) \mid 0 \leq i_j < s_j, j = 1, \ldots, m\}$ constitutes a basis for $\mathbf{F}_q^n$ as a vectorspace over $\mathbf{F}_q$. For this purpose it is sufficient to show that the restriction of ev to

$$\{G(X_1, \ldots, X_m) \mid \deg_{X_i}(G) < s_i, i = 1, \ldots, m\} \tag{1}$$

is surjective. Given $(a_1, \ldots, a_n) \in \mathbf{F}_q^n$ let

$$F(X_1, \ldots, X_m) = \sum_{v=1}^{n} a_v \prod_{i=1}^{m} \prod_{a \in \mathbf{F}_q \setminus \{P_i^{(v)}\}} \left(\frac{X_i - a}{P_i^{(v)} - a} \right).$$

Here, we have used the notation $P_v = (P_1^{(v)}, \ldots, P_n^{(v)})$, $v = 1, \ldots, n$. It is clear that $\mathrm{ev}_S(F) = (a_1, \ldots, a_n)$ and therefore $\mathrm{ev}_S : \mathbf{F}_q[X_1, \ldots, X_m] \to \mathbf{F}_q^n$ is surjective. Consider a monomial ordering. Let $R(X_1, \ldots, X_m)$ be the remainder of $F(X_1, \ldots, X_m)$ after division with $\{A_1(X_1, \ldots, X_m), \ldots, A_m(X_1, \ldots, X_m)\}$. Here, $A_i(X_1, \ldots, X_m) = \prod_{a \in S_i}(X_i - a)$, $i = 1, \ldots, m$. Clearly, $F(P_i) = R(P_i) = a_i$, $i = 1, \ldots, n$. Hence, the restriction of ev_S to (1) is indeed surjective.

We next show how to estimate the minimum distance of $E(\mathbb{M}, S)$. The Schwartz-Zippel bound [15, 17, 4] is as follows:

Theorem 1. *Let $X_1^{i_1} \cdots X_m^{i_m}$ be the leading monomial of $F(X_1, \ldots, X_m)$ with respect to a lexicographic ordering. The number of elements in $S = S_1 \times \cdots \times S_m$ that are zeros of F is at most $i_1 s_2 \cdots s_m + s_1 i_2 s_3 \cdots s_m + \cdots + s_1 \cdots s_{m-1} i_m$.*

The proof of this result is purely combinatorial. Using the inclusion-exclusion principle it can actually be strengthened to the following result which is a special case of the footprint bound from Gröbner basis theory:

Theorem 2. *Let notation be as in Theorem 1. The number of elements in S that are zeros of F is at most equal to $n - (s_1 - i_1)(s_2 - i_2) \cdots (s_m - i_m)$.*

Proposition 2. *The minimum distance of $E(\mathbb{M}, S)$ is at least*

$$\min\{(s_1 - i_1)(s_2 - i_2) \cdots (s_m - i_m) | X_1^{i_1} \cdots X_m^{i_m} \in \mathbb{M}\}.$$

The bound is sharp if for every $M \in \mathbb{M}$ all divisors of M also belong to $\mathbb{M}$.

Proof. The first part follows from Theorem 2. To see the last part write for $i = 1, \ldots, m$, $S_i = \{b_1^{(i)}, \ldots, b_{|S_i|}^{(i)}\}$. Clearly, $F(X_1, \ldots X_m) = \prod_{v=1}^{m} \prod_{j=1}^{i_v} \left(X_v - b_j^{(v)}\right)$ has leading monomial $X_1^{i_1} \cdots X_m^{i_m}$ with respect to any monomial ordering and evaluates to zero in exactly $n - (s_1 - i_1)(s_2 - i_2) \cdots (s_m - i_m)$ points from S. Finally, any monomial that occurs in the support of F is a factor of $X_1^{i_1} \cdots X_m^{i_m}$.

3 Bounding the number of zeros of multiplicity r

In the following let $\boldsymbol{X} = (X_1, \ldots, X_m)$ and $\boldsymbol{T} = (T_1, \ldots, T_m)$.

Definition 1. *Let $\mathbf{F}$ be any field. Given $F(\boldsymbol{X}) \in \mathbf{F}[\boldsymbol{X}]$ and $\boldsymbol{k} \in \mathbf{N}_0^m$ then the $\boldsymbol{k}$'th Hasse derivative of F, denoted by $F^{(\boldsymbol{k})}(\boldsymbol{X})$, is the coefficient of $\boldsymbol{T}^{\boldsymbol{k}}$ in $F(\boldsymbol{X} + \boldsymbol{T})$. In other words $F(\boldsymbol{X} + \boldsymbol{T}) = \sum_{\boldsymbol{k}} F^{(\boldsymbol{k})}(\boldsymbol{X}) \boldsymbol{T}^{\boldsymbol{k}}$.*

The concept of multiplicity for univariate polynomials is generalized to multivariate polynomials in the following way:

Definition 2. *For $F(\boldsymbol{X}) \in \mathbf{F}[\boldsymbol{X}]\backslash\{0\}$ and $\boldsymbol{a} \in \mathbf{F}^m$ we define the multiplicity of F at $\boldsymbol{a}$ denoted by mult$(F, \boldsymbol{a})$ as follows: Let r be an integer such that for every $\boldsymbol{k} = (k_1, \ldots, k_m) \in \mathbf{N}_0^m$ with $k_1 + \cdots + k_m < r$, $F^{(\boldsymbol{k})}(\boldsymbol{a}) = 0$ holds, but for some $\boldsymbol{k} = (k_1, \ldots, k_m) \in \mathbf{N}_0^m$ with $k_1 + \cdots + k_m = r$, $F^{(\boldsymbol{k})}(\boldsymbol{a}) \neq 0$ holds, then mult$(F, \boldsymbol{a}) = r$. If $F = 0$ then we define mult$(F, \boldsymbol{a}) = \infty$.*

Elaborating on the results in [5] we find:

Theorem 3. *Let $F(\boldsymbol{X}) \in \mathbf{F}[\boldsymbol{X}]$ be a non-zero polynomial and let $X_1^{i_1} \cdots X_m^{i_m}$ be its leading monomial with respect to a lexicographic ordering. Then for any finite sets $S_1, \ldots, S_m \subseteq \mathbf{F}$*

$$\sum_{\boldsymbol{a} \in S_1 \times \cdots \times S_m} \text{mult}(F, \boldsymbol{a}) \leq i_1 s_2 \cdots s_m + s_1 i_2 s_3 \cdots s_m + \cdots + s_1 \cdots s_{m-1} i_m.$$

Theorem 1 generalizes to Corollary 1 which we call the Schwartz-Zippel bound.

Corollary 1. *Let $F(\boldsymbol{X}) \in \mathbf{F}[\boldsymbol{X}]$ be a non-zero polynomial and let $X_1^{i_1} \cdots X_m^{i_m}$ be its leading monomial with respect to a lexicographic ordering. Assume $S_1, \ldots, S_m \subseteq \mathbf{F}$ are finite sets. Then over $S_1 \times \cdots \times S_m$ the number of zeros of multiplicity at least r is less than or equal to*

$$\big(i_1 s_2 \cdots s_m + s_1 i_2 s_3 \cdots s_m + \cdots + s_1 \cdots s_{m-1} i_m\big)/r. \tag{2}$$

Just as we were able to improve upon Theorem 1 by using the inclusion-exclusion principle we will also be able to improve upon Corollary 1. However, now the situation is much more complex and therefore the inclusion-exclusion principle is no longer sufficient. What we need to strengthen Corollary 1 is the following rather technical function:

Definition 3. *Let $r \in \mathbf{N}, i_1, \ldots, i_m \in \mathbf{N}_0$. Define $D(i_1, r, s_1) = \min\big\{\lfloor \frac{i_1}{r} \rfloor, s_1\big\}$ and for $m \geq 2$*

$$\begin{aligned} &D(i_1, \ldots, i_m, r, s_1, \ldots, s_m) = \\ &\max_{(u_1,\ldots,u_r) \in A(i_m, r, s_m)} \Big\{ (s_m - u_1 - \cdots - u_r) D(i_1, \ldots, i_{m-1}, r, s_1, \ldots, s_{m-1}) \\ &\qquad + u_1 D(i_1, \ldots, i_{m-1}, r-1, s_1, \ldots, s_{m-1}) + \cdots \\ &\qquad + u_{r-1} D(i_1, \ldots, i_{m-1}, 1, s_1, \ldots, s_{m-1}) + u_r s_1 \cdots s_{m-1} \Big\} \end{aligned}$$

where

$$\begin{aligned} &A(i_m, r, s_m) = \\ &\{(u_1, \ldots, u_r) \in \mathbf{N}_0^r \mid u_1 + \cdots + u_r \leq s_m \quad \text{and} \quad u_1 + 2u_2 + \cdots + r u_r \leq i_m\}. \end{aligned}$$

Theorem 4. *For a polynomial $F(\boldsymbol{X}) \in \mathbf{F}[\boldsymbol{X}]$ let $X_1^{i_1} \cdots X_m^{i_m}$ be its leading monomial with respect to the lexicographic ordering with $X_m \prec \cdots \prec X_1$. Then F has at most $D(i_1, \ldots, i_m, r, s_1, \ldots, s_m)$ zeros of multiplicity at least r in $S_1 \times \cdots \times S_m$. The corresponding recursive algorithm produces a number that is at most equal to the number found in Corollary 1 and at most equal to $s_1 \cdots s_m$.*

Proof. The proof involves a modification of the method described in [5].

Remark 1. Given $(i_1, \ldots, i_m, r, s_1, \ldots, s_m)$ with $\lfloor i_1/s_1 \rfloor + \cdots + \lfloor i_m/s_m \rfloor \geq r$ then there exist polynomials with the leading monomial being $X_1^{i_1} \cdots X_m^{i_m}$ such that all points in $S_1 \times \cdots \times S_m$ are zeros of multiplicity at least r. Hence, we need only apply the algorithm to tuples $(i_1, \ldots, i_m)$ such that

$$\lfloor i_1/s_1 \rfloor + \cdots + \lfloor i_m/s_m \rfloor < r. \tag{3}$$

In a number of experiments listed in [8] we calculated $D(i_1, \ldots, i_m, r, q, \ldots, q)$ for various choices of m, q and r and for all values of $(i_1, \ldots, i_m)$ satisfying (3). We here list the maximal attained improvement obtained by using Theorem 4 rather than using Corollary 1. We do this relatively to the number of points q^m. In other words we list in Table 1 the value

$$\left(\max_{i_1, \ldots, i_m} \{ \min\{(i_1 + \cdots i_m) q^{m-1}/r, q^m\} - D(i_1, \ldots, i_m, r, q, \ldots, q) \} \right) / q^m$$

for various choices of m, q, r. The experiments also show distinct average improvement. This is illustrated in Table 2 where for fixed q, r, m we list the mean value of

$$\frac{\min\{(i_1 + \cdots + i_m) q^{m-1}, q^m\} - D(i_1, \ldots, i_m, r, q, \ldots, q)}{\min\{(i_1 + \cdots + i_m) q^{m-1}, q^m\}}. \tag{4}$$

The average is taken over the set of exponents $(i_1, \ldots, i_m) \neq \mathbf{0}$ where $\lfloor i_1/q \rfloor + \cdots + \lfloor i_m/q \rfloor < r$ holds.

Table 1. Maximum improvements relative to q^m; truncated

m		2				3				4	
r		2	3	4	5	2	3	4	5	2	3
	2	0.25	0.25	0.25	0.25	0.25	0.375	0.375	0.375	0.312	0.375
	3	0.222	0.222	0.222	0.222	0.296	0.296	0.296	0.296	0.296	0.333
	4	0.187	0.187	0.187	0.187	0.281	0.25	0.25	0.265	0.316	0.289
q	5	0.24	0.16	0.16	0.2	0.256	0.256	0.232	0.24	0.307	0.288
	7	0.204	0.204	0.163	0.142	0.279	0.244	0.227	0.209	0.299	0.276
	8	0.234	0.203	0.171	0.140	0.275	0.25	0.214	0.203	0.299	0.275

The values $D(i_1, \ldots, i_m, r, s_1, \ldots, s_m)$ may sometimes be time consuming to calculate. Therefore it is relevant to have some closed formula estimates of these numbers. We next present such estimates for the case of two variables. By Remark 1 the following proposition covers all non-trivial cases:

Table 2. The mean value of (4); truncated

m		2				3				4	
r		2	3	4	5	2	3	4	5	2	3
	2	0.363	0.273	0.337	0.291	0.301	0.300	0.342	0.307	0.248	0.260
	3	0.217	0.286	0.228	0.236	0.194	0.224	0.213	0.214	0.158	0.177
	4	0.191	0.197	0.232	0.195	0.158	0.169	0.180	0.172	0.125	0.135
q	5	0.155	0.167	0.174	0.197	0.139	0.145	0.148	0.153	0.110	0.116
	7	0.128	0.137	0.138	0.138	0.119	0.122	0.121	0.119	0.093	0.098
	8	0.126	0.127	0.134	0.126	0.114	0.115	0.113	0.111	0.089	0.093

Proposition 3. *For $k = 1, \ldots, r-1$, $D(i_1, i_2, r, s_1, s_2)$ is upper bounded by*

(C.1) $s_2\frac{i_1}{r} + \frac{i_2}{r}\frac{i_1}{r-k}$
if $(r-k)\frac{r}{r+1}s_1 \leq i_1 < (r-k)s_1$ *and* $0 \leq i_2 < ks_2$,

(C.2) $s_2\frac{i_1}{r} + ((k+1)s_2 - i_2)(\frac{i_1}{r-k} - \frac{i_1}{r}) + (i_2 - ks_2)(s_1 - \frac{i_1}{r})$
if $(r-k)\frac{r}{r+1}s_1 \leq i_1 < (r-k)s_1$ *and* $ks_2 \leq i_2 < (k+1)s_2$,

(C.3) $s_2\frac{i_1}{r} + \frac{i_2}{k+1}(s_1 - \frac{i_1}{r})$
if $(r-k-1)s_1 \leq i_1 < (r-k)\frac{r}{r+1}s_1$ *and* $0 \leq i_2 < (k+1)s_2$.

Finally,

(C.4) $D(i_1, i_2, r, s_1, s_2) = s_2\lfloor\frac{i_1}{r}\rfloor + i_2(s_1 - \lfloor\frac{i_1}{r}\rfloor)$
if $s_1(r-1) \leq i_1 < s_1 r$ *and* $0 \leq i_2 < s_2$.

The above numbers are at most equal to $\min\{(i_1 s_2 + s_1 i_2)/r, s_1 s_2\}$.

Proof. The estimates are found by treating i_1 and i_2 as rational numbers.

4 The decoding algorithm

The main ingredient of the algorithm is to find an interpolation polynomial

$$Q(X_1, \ldots, X_m, Z) = Q_0(X_1, \ldots, X_m) + Q_1(X_1, \ldots, X_m)Z + \cdots + Q_t(X_1, \ldots, X_m)Z^t$$

such that $Q(X_1, \ldots, X_m, F(X_1, \ldots, X_m))$ cannot have more than $n - E$ different zeros of multiplicity at least r whenever $\mathrm{Supp}(F) \subseteq \mathbb{M}$. The integer E above is the number of errors to be corrected by our list decoding algorithm. To fulfill this requirement we will define appropriate sets of monomials $B(i, E, r)$, $i = 1, \ldots, t$ and then require $Q_i(X_1, \ldots, X_m)$ to be chosen such that $\mathrm{Supp}(Q_i) \subseteq B(i, E, r)$. Rather than using the results from the previous section on all possible choices of $F(X_1, \ldots, X_m)$ with $\mathrm{Supp}(F) \subseteq \mathbb{M}$ we only consider the worst cases where the leading monomial of F is contained in the following set:

Definition 4. $\overline{\mathbb{M}} = \{M \in \mathbb{M} \mid$ *if* $N \in \mathbb{M}$ *and* $M|N$ *then* $M = N\}$.

Hence, $\overline{\mathbb{M}}$ is so to speak the boarder of $\mathbb{M}$.

Definition 5. *Given positive integers i, E, r with $E < n$ let*

$$B(i,E,r) = \{K \in \Delta(r,m) \mid D_r(KM^i) < n - E \ \text{ for all } \ M \in \overline{\mathbb{M}}\}.$$

Here, $D_r(X_1^{i_1}, \ldots, X_m^{i_m})$ can either be $D(i_1, \ldots, i_m, r, s_1, \ldots, s_m)$ or in the case of two variables it can be the numbers from Proposition 3. Another option would be to let $D_r(X_1^{i_1}, \ldots, X_m^{i_m})$ be the number in (2). With a reference to Remark 1 we have defined $\Delta(r,m) = \{X_1^{i_1} \cdots X_m^{i_m} \mid \lfloor i_1/s_1 \rfloor + \cdots + \lfloor i_m/s_m \rfloor < r\}$.

The decoding algorithm calls for positive integers t, E, r such that

$$\sum_{i=1}^{t} |B(i,E,r)| > nN(m,r), \tag{5}$$

where $N(m,r) = \binom{m+r}{m+1}$ is the number of linear equations to be satisfied for a point in $\mathbf{F}_q^{m+1}$ to be a zero of $Q(X_1, \ldots, X_m, Z)$ of multiplicity at least r. As we will see condition (5) ensures that we can correct E errors. We will say that (t, E, r) satisfies the initial condition if given the pair (E, r), t is the smallest integer such that (5) is satisfied. Whenever this is the case we let $B'(t,E,r)$ be any subset of $B(t,E,r)$ such that $\sum_{i=1}^{t-1} |B(i,E,r)| + |B'(t,E,r)| = nN(m,r)+1$. Replacing $B(t,E,r)$ with $B'(t,E,r)$ lowers the run time of the algorithm.

Algorithm 1 *Input: Received word $\boldsymbol{r} = (r_1, \ldots, r_n) \in \mathbf{F}_q^n$. Set of integers (t, E, r) that satisfies the initial condition and corresponding sets $B(1,E,r) \ldots, B(t-1,E,r), B'(t,E,r)$.*

Step 1 Find non-zero polynomial

$$Q(X_1, \ldots, X_m Z) = Q_0(X_1, \ldots, X_m) + Q_1(X_1, \ldots, X_m)Z + \cdots + Q_t(X_1, \ldots, X_m)Z^t$$

such that

1. *$Supp(Q_i) \subseteq B(i,E,r)$ for $i = 1, \ldots, t-1$ and $Supp(Q_t) \subseteq B'(t, EP, r)$,*
2. *(P_i, r_i) is a zero of $Q(X_1, \ldots, X_m, Z)$ of multiplicity at least r, $i = 1, \ldots, n$.*

Step 2 Find all $F(X_1, \ldots, X_m) \in \mathbf{F}_q[X_1, \ldots, X_m]$ such that

$$(Z - F(X_1, \ldots, X_m)) | Q(X_1, \ldots, X_m, Z). \tag{6}$$

Output: A list containing $(F(P_1), \ldots, F(P_n))$ for all F satisfying (6).

Theorem 5. *The output of Algorithm 1 contains all words in $E(\mathbb{M}, S)$ within distance E from the received word $\boldsymbol{r}$. Once the preparation step has been performed the algorithm runs in time $\mathcal{O}(\bar{n}^3)$ where $\bar{n} = n\binom{m+r}{m+1}$. For given multiplicity r the maximal number of correctable errors E and the corresponding sets $B(1,E,r), \ldots, B(t-1,E,r)$, $B'(t,E,r)$ can be found in time $\mathcal{O}(n \log(n) r^m s' |\overline{\mathbb{M}}|/\sigma)$ assuming that the values of the function D_r are known. Here $\sigma = \max\{i_1 + \cdots + i_m | X_1^{i_1} \cdots X_m^{i_m} \in \overline{\mathbb{M}}\}$ and $s' = \max\{s_1, \ldots, s_m\}$.*

Proof. The interpolation problem corresponds to $\bar{n}$ homogeneous linear equations in $\bar{n}+1$ unknowns. Hence, indeed a suitable Q can be found in time $\mathcal{O}(\bar{n}^3)$. Now assume $\mathrm{Supp}(F) \subseteq \mathbb{M}$ and that $\mathrm{dist}_H(\mathrm{ev}_S(F), \boldsymbol{r}) \leq E$. Then P_j is a zero of $Q(X_1, \ldots, X_m, F(X_1, \ldots, X_m))$ of multiplicity at least r for at least $n - E$ choices of j. By the definition of $B(i, E, r)$ this can, however, only be the case if $Q(X_1, \ldots, X_m, F(X_1, \ldots, X_m)) = 0$. Therefore, $Z - F(X_1, \ldots, X_m)$ is a factor in $Q(X_1, \ldots, X_m, Z)$. Finding linear factors of polynomials in $(\mathbf{F}_q[X_1, \ldots, X_m])[Z]$ can be done in time $\mathcal{O}(\bar{n}^3)$ by applying Wu's algorithm in [16] (see [13, p. 20]).

5 Examples

Example 1. In this example we consider a point ensemble $S = S_1 \times S_2$ with $s_1 = 128$ and $s_2 = 64$. We first consider codes $E(\mathbb{M}, S)$ where $\mathbb{M} = \{X_1^{i_1} X_2^{i_2} \mid i_1 + 2i_2 \leq u\}$. Note, the weight 2 which corresponds to the number s_1/s_2. The performance of Algorithm 1 is independent of the field in which S_1 and S_2 live. In Table 3 we list the number of errors that we can correct when the function $D_r(i_1, i_2)$ – is chosen to be $D(i_1, i_2, r, 80, 80)$ (column D), – is the closed formula expression from Proposition 3 (column C), – or is the Schwartz-Zippel bound (column S), respectively. The row $\lfloor (d-1)/2 \rfloor$ corresponds to half the minimum distance and the row Dim. is the dimension of the code. We next consider codes where $\mathbb{M} = \{X_1^{i_1} X_2^{i_2} \mid i_1 < k_1, i_2 < k_2\}$. That is, we consider Reed-Solomon product codes. In Table 4 we list the number of errors that can be corrected by Algorithm 1.

Example 2. In this example we consider a point ensemble $S = S_1 \times S_2$ with $s_1 = s_2 = 80$. We consider codes $E(\mathbb{M}, S)$ where $\mathbb{M} = \{X_1^{i_1} X_2^{i_2} \mid i_1 + i_2 \leq u\}$ for various values of u. That is, we consider Reed-Muller like codes. The performance of Algorithm 1 is independent on the field in which S_1 and S_2 live. In Table 5 we list the number of errors that we can correct when the function $D_r(i_1, i_2)$ – is chosen to be $D(i_1, i_2, r, 80, 80)$ (column D), – is the closed formula expression from Proposition 3 (column C), – or is the Schwartz-Zippel bound (column S), respectively. Column A corresponds to a bound from [2] on what can be achieved by applying their algorithm. The row A_∞ corresponds to what could theoretically be achieved by the algorithm in [2] if one uses high enough multiplicity.
Assuming $S_1, S_2 \subseteq \mathbf{F}_{128}$, $E(\mathbb{M}, S)$ corresponds to a puncturing of the generalized Reed-Muller code $\mathrm{RM}_{128}(u, 2)$. This suggest that as an alternative to using Algorithm 1 one could decode with respect to $\mathrm{RM}_{128}(u, 2)$ treating the punctured points as errors. The best known algorithm to decode $\mathrm{RM}_{128}(u, 2)$ for the values of u considered in this example is the algorithm by Pellikaan and Wu which uses the subfield subcode approach. Row PW of the table explains what can be achieved by this alternative approach.

6 Concluding remarks

In this work we presented a decoding algorithm that works for a large class of affine variety codes. By a number of experiments we illustrated the usefulnes of our algorithm. The work was supported in part by Danish Natural Research Council grant 272-07-0266. The authors thank Peter Beelen and Teo Mora for pleasant discussions. Also thanks to L. Grubbe Nielsen for linguistical assistance.

Table 3. Error correction capabilities for the first codes in Example 1

u	3			4			7			20		
r	D	C	S	D	C	S	D	C	S	D	C	S
2	5129	5105	4895	4799	4777	4575	4143	4124	3871	2487	2475	2175
3	5367	5333	5205	5048	5016	4906	4407	4381	4245	2855	2833	2666
4	5474	5438	5343	5180	5143	5071	4566	4535	4431	3060	3031	2927
9		5653	5617		5390	5361		4817	4785		3415	3384
20		5757	5740		5509	5494		4959	4943		3609	3599
$\lfloor \frac{d-1}{2} \rfloor$		3999			3967			3871			3455	
Dim.		6			9			20			121	

Table 4. Error correction capabilities for the second codes in Example 1.

(k_1, k_2)	$(4,7)$			$(5,9)$			$(8,15)$			$(21,41)$		
r	D	C	S	D	C	S	D	C	S	D	C	S
2	4036	4015	3519	3655	3639	3071	2820	2808	2111	1061	1055	0
3	4289	4261	3903	3911	3885	3498	3077	3058	2602	1183	1171	533
4	4411	4381	4111	4042	4011	3727	3214	3187	2847	1310	1291	831
9		4598	4487		4244	4124		3455	3313		1567	1365
20		4711	4662		4364	4310		3588	3526		1704	1615
$\lfloor \frac{d-1}{2} \rfloor$		3720			3599			3248			1935	
Dim.		28			45			120			861	

Table 5. Error correction capabilities for the codes in Example 2

u	3				4				7				20			
r	D	C	S	A	D	C	S	A	D	C	S	A	D	C	S	A
2	3594	3571	3399	3310	3317	3297	3119	2999	2693	2679	2479	2302	1279	1279	999	585
3	3791	3765	3679	3604	3524	3499	3413	3323	2943	2918	2799	2692	1575	1559	1439	1138
4	3899	3869	3799	3758	3647	3618	3559	3492	3080	3058	2979	2896		1728	1639	1428
9		4072	4053	4027		3837	3813	3792		3315	3297	3253		2053	2035	1935
20		4171	4163	4152		3946	3939	3926		3444	3435	3418		2219	2211	2169
A_∞		4257				4042				3558				2368		
PW		3891				3503				2568				0		
$\lfloor \frac{d-1}{2} \rfloor$		3079				3039				2919				2399		
Dim.		10				15				36				231		

References

1. D. Augot, M. El-Khamy, R. J. McEliece, F. Parvaresh, M. Stepanov, and A. Vardy, "List decoding of Reed-Solomon product codes," in *Proceedings of the Tenth International Workshop on Algebraic and Combinatorial Coding Theory, Zvenigorod, Russia,*, Sept. 2006, pp. 210-213.
2. D. Augot and M. Stepanov, "A Note on the Generalisation of the Guruswami-Sudan List Decoding Algorithm to Reed-Muller Codes," in *Gröbner Bases, Coding, and Cryptography,* Springer 2009, Eds. Sala, Mora, Perret, Sakata, and Traverso, pp. 395-398.
3. M. Bras-Amors and M. E. O'Sullivan, "Duality for some families of correction capability optimized evaluation codes," *Adv. in Math. of Comm.*, **2**, 2008, pp. 15-33.
4. R. A. DeMillo and R. J. Lipton, "A Probabilistic Remark on Algebraic Program Testing," *Information Processing Letters,* **7**, no. 4, June 1978, pp. 193-195.
5. Z. Dvir, S. Kopparty, S. Saraf, M. Sudan, "Extensions to the Method of Multiplicities, with applications to Kakeya Sets and Mergers," (appeared in Proc. of FOCS 2009) arXiv:0901.2529v2, 2009, 26 pages.
6. O. Geil and T. Høholdt, On Hyperbolic Codes, Proc. AAECC-14, *Lecture Notes in Comput. Sci. 2227*, 2001, pp. 159-171
7. O. Geil and R. Matsumoto, "Generalized Sudan's list decoding for order domain codes," Proc. AAECC-16, *Lecture Notes in Comput. Sci., 4851*, Springer, 2007, pp. 50-59.
8. O. Geil and C. Thomsen, "Tables for numbers of zeros with multiplicity at least r," webpage: `http://zeros.spag.dk`, January 18th, 2011.
9. V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inform. Theory*, **45**, 1999, pp. 1757-1767.
10. J. P. Hansen, "Toric Varieties Hirzebruch Surfaces and Error-Correcting Codes," *Appl. Alg. in Eng. Comm. and Comp.*, **13**, 2002, pp. 289-300.
11. D. Joyner, "Toric Codes over Finite Fields," *Appl. Alg. in Eng. Comm. and Comp.*, **15**, 2004, pp. 63-79.
12. R. Pellikaan and X.-W. Wu, "List Decoding of q-ary Reed-Muller Codes," *IEEE Trans. Inform. Theory*, **50**, 2004, pp. 679-682.
13. R. Pellikaan and X.-W. Wu, "List Decoding of q-ary Reed-Muller Codes," (Expanded version of the paper [12]), available from `http://win.tue.nl/~ruudp/paper/43-exp.pdf`, 37 pages.
14. D. Ruano: "On the structure of generalized toric codes," *Journal of Symb. Comp.*, **44**, 2009, pp. 499-506.
15. J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. Assoc. Comput. Mach.* , **27**, no. 4, 1980, pp. 701–717.
16. X.-W. Wu, "An Algorithm for Finding the Roots of the Polynomials over Order Domains," in Proc. of 2002, *IEEE Int. Symp. on Inf. Th.*, Lausanne, June 2002.
17. R. Zippel, "Probabilistic algorithms for sparse polynomials," Proc. of EUROSAM 1979, *Lect. Notes in Comp. Sc.,* **72**, Springer, Berlin, 1979, pp. 216–226.