CS 250 FINAL PROJECT
MULTIPLICITY CODES

KATHARINE WOO AND PATRICK REVILLA

1. Introduction and motivation

In classical error-correcting codes, we aim to build encoding schemes $\mathcal{C}$, such that a message $\mathbf{x}$ can be recovered even if $\mathcal{C}(\mathbf{x})$ is corrupted in a few coordinates. We saw that Reed-Solomon codes were optimal in many ways for error correction; these codes meet the Singleton bound, are MDS codes and can be decoded efficiently using the Berlekamp-Welch algorithm. However, sometimes decoding all of $\mathbf{x}$ is more information than we desire.

Consider the following situation: Alice encoded her hard drive and saved its current state. A year later, her encoded hard drive has been corrupted in a few spots. Alice wants to access only one of her files from the hard drive. In this scenario, using a error correcting code and fully decoding the whole hard drive is overkill. Instead, she wants to be able to perform local correction and decoding. We recall that local correction is the ability to decode a particular part of the message more efficiently. More formally, we have the following definition.

Definition 1. A code $\mathcal{C} \subset \mathbb{F}_q^n$ is (δ, Q, γ) **locally correctable** if there is a randomized algorithm A such that for all $w \in \mathbb{F}_q^n$ such that there exists some $c \in \mathcal{C}$ where $\delta(w, c) < \delta$, for each $i \in [n]$,

$$\Pr[A^w(i) = c_i] \geq 1 - \gamma.$$

Here, A^w has oracle access to w and makes at most Q queries.

We saw that Reed Solomon codes are unfortunately not locally correctable. Luckily, a variation of Reed Solomon codes, Reed Muller codes, are locally correctable.

Definition 2. The m-variate **Reed Muller code** of degree r over $\mathbb{F}_q$ is

$$\mathrm{RM}_q(m, r) = \{(f(\alpha_1), ..., f(\alpha_m)) : f \in \mathbb{F}_q[x_1, ..., x_m], \deg(f) \leq r\}$$

For Reed Muller codes, we have the following rate and query data.

Q	n	Code
2	$\Theta(2^k)$	$\mathrm{RM}_2(2, 1)$
$\log(n)$	$\mathrm{poly}(k)$	$\mathrm{RM}_q(m, r)$ for $m \approx q/\log(q)$
n^ϵ	$\Theta_\epsilon(k)$	$\mathrm{RM}_q(m, r)$ for $m = 1/\epsilon$
$\sqrt{n}$	$n = 8k$	$\mathrm{RM}_2(2, q/2)$

We can see that the rate of Reed Muller codes is very low. So, we aim to create codes that are locally correctable with better rate.

Date: March 2019.

1.1. **A toy model.** We will discuss a natural variation of Reed Muller codes. We recall that for polynomial evaluation it sufficed to consider polynomials in $\mathbb{F}_q[X]$ with degree up to q because X^q acts the same way as X. However, when we add derivatives, we see that $qX^{q-1} \equiv 0$ in $\mathbb{F}_q[X]$ which is not the same as 1 in $\mathbb{F}_q[X]$. So, if we add derivatives, we can now consider polynomials of higher degree and distinguish them. These properties motivate why it makes sense to add derivatives to our Reed Muller codes and see what new features we gain. For this model, we will consider only first derivatives and bivariate polynomials. This model is discussed by Kopparty, Saraf and Yekhanin in [4].

Definition 3. Our bivariate toy code of degree d over $\mathbb{F}_q$ is defined as

$$\mathcal{C}_{toy} = \left\{ \left(P(\mathbf{a}), \frac{\partial P}{\partial X}(\mathbf{a}), \frac{\partial P}{\partial Y}(\mathbf{a}) \right)_{\mathbf{a} \in \mathbb{F}_q^2} \in (\mathbb{F}_q^3)^{q^2} : P \in \mathbb{F}_q[X,Y], \deg(P) \leq d \right\}$$

Here it is important to note that the toy code is taken over $(\mathbb{F}_q^3)^{q^2}$. For notation purposes, we will denote

$$\operatorname{Enc}(P) := \left(P(\mathbf{a}), \frac{\partial P}{\partial X}(\mathbf{a}), \frac{\partial P}{\partial Y}(\mathbf{a}) \right)_{\mathbf{a} \in \mathbb{F}_q^2}$$

Now we will discuss the basic properties of this code.

(1) Distance: For polynomial P to give a codeword that is 0 at spot $\mathbf{a}$, we have that $P(\mathbf{a}) = \frac{\partial P}{\partial X}(\mathbf{a}) = \frac{\partial P}{\partial Y}(\mathbf{a}) = 0$. In other words, P has a root at $\mathbf{a}$ with multiplicity at least 2. A polynomial with degree d in two variables can have at most $\frac{dq}{2}$ zeros of multiplicity 2 and this bound can be attained. This also follows from the more general statement of Lemma 2.1. Thus, the relative distance is $\delta = 1 - \frac{d}{2q}$.

(2) Rate: we have $n = 3q^2$ in this scenario. Then we can see that k is the number of polynomials of $\mathbb{F}_q[X,Y]$ with degree $\leq d$. Using a counting argument, we have that this is $\binom{d+2}{2}$. So, $R = \frac{\binom{d+2}{2}}{3q^2} \approx \frac{2(1-\delta)^2}{3}$

Remark. We note that the distance argument and rate argument here are corollaries of the more general Theorem 2.2 proven in §2. Even in this toy case, the rate of our toy code is asymptotically better than that of bivariate Reed Muller codes. Now we want to discuss the local correctability of this code.

1.1.1. *Local correction for $\mathcal{C}_{toy}$.* We give a rough sketch of the local correction algorithm for $\mathcal{C}_{toy}$ to motivate the local correction algorithm described in §4.1. We direct the reader to look towards §4.1 for a detailed analysis of the algorithm.

Local Correction Algorithm for $\mathcal{C}_{toy}$:

Input: $r \in (\mathbb{F}_q^3)^{q^2}$ such that $\delta(r, \operatorname{Enc}(P))$ is small for some P, and some $\mathbf{a} \in \mathbb{F}_q^2$.

Output: $(P(\mathbf{a}), \frac{\partial P}{\partial X}(\mathbf{a}), \frac{\partial P}{\partial Y}(\mathbf{a}))$

(1) **Pick random directions**: Choose $\mathbf{b}_1, \mathbf{b}_2 \overset{R}{\leftarrow} \mathbb{F}_q^2$. We will denote $\mathbf{b}_1 = (b_{1,X}, b_{1,Y})$ and $\mathbf{b}_2 = (b_{2,X}, b_{2,Y})$.

(2) **Solve for $Q_\mathbf{b}(T)$ for each $\mathbf{b}$**: Let $L_\mathbf{b} = \{\mathbf{a} + \mathbf{b}T : T \in \mathbb{F}_q\}$ be a line. We want to look at r on $L_\mathbf{b}$ which are q noisy evaluations of $Q_\mathbf{b}(T)$ and $\frac{\partial Q_\mathbf{b}}{\partial T}$ where $\deg(Q_\mathbf{b}) \leq d$. This is enough to recover a polynomial $Q_\mathbf{b}(T)$.

(3) **Solve a system of equations**: We want to solve for x, y that satisfy the following linear system of equations. With high probability $\mathbf{b}_1, \mathbf{b}_2$ will be linearly independent and the system will be solvable.

$$b_{1,X}x + b_{1,Y}y = \frac{\partial Q_{\mathbf{b}_1}}{\partial T}(0),$$
$$b_{2,X}x + b_{2,Y}y = \frac{\partial Q_{\mathbf{b}_2}}{\partial T}(0).$$

If there is a unique solution x, y given, then we proceed. Otherwise we output FAIL. If $Q_{\mathbf{b}_1}(0) \neq Q_{\mathbf{b}_2}(0)$, we output FAIL. Otherwise we set this quantity to be p.

(4) Output (p, x, y).

In step (2) with high probability of the choice of $\mathbf{b}$, $r \mid_{L_\mathbf{b}}$ and $\mathrm{Enc}(P) \mid_{L_\mathbf{b}}$ agree at many places, so $Q_\mathbf{b}(T) = P(\mathbf{a} + \mathbf{b}T)$. We saw that $Q_\mathbf{b}(T) = P(\mathbf{a} + \mathbf{b}T)$. Now we note that

$$\left(Q_\mathbf{b}(0), \frac{\partial Q_\mathbf{b}}{\partial T}(0)\right) = \left(P(\mathbf{a}), b_X \frac{\partial P}{\partial X}(\mathbf{a}) + b_Y \frac{\partial P}{\partial Y}(\mathbf{a})\right).$$

So, if r and $\mathrm{Enc}(P)$ are close enough together and we choose reasonably good $\mathbf{b}_1, \mathbf{b}_2$, we will have that $p = P(\mathbf{a})$, $x = \frac{\partial P}{\partial X}(\mathbf{a})$ and $y = \frac{\partial P}{\partial Y}(\mathbf{a})$ as desired. In this algorithm, r is queried $2q$ times.

This toy case shows us that adding variables and derivatives will likely allow us to increase the rate while keeping the number of queries not too large. In particular, multiplicity codes which are similar to the toy case but with more variables and more derivatives will give explicit codes for the following theorem shown by Kopparty, Saraf, and Yekhanin in [4].

Theorem 1.1. *For every $0 < \epsilon, \alpha < 1$, for infinitely many n, there is a code $\mathcal{C}$ over an alphabet Σ where $|\Sigma| \leq n^{O(1)}$ such that $\mathcal{C}$ has length n, rate at least $1 - \alpha$, distance $\delta \geq \epsilon\alpha/2$ and is locally self-correctable from $\delta/10$ errors with $O(n^\epsilon)$ queries.*

To get to this result we will discuss multiplicity codes in more detail. We introduce the code and its basic properties in §2. In §3, we give a global decoding algorithm for these codes. In §4 we will describe the local correction algorithm, and in §5 we discuss variations on the way to efficiently encode multiplicity codes. Finally we discuss in §6 remaining questions about multiplicity codes.

2. Multiplicity codes

To discuss multiplicity codes, we must first introduce some notation. For the vector $\mathbf{i} = \langle i_1, ..., i_m \rangle$ of non-negative integers, we define **the weight of i** as

$$\mathrm{wt}\,\mathbf{i} := \sum_j i_j,$$

the total sum of its entries (we note that this is different from the in-class definition of weight). Additionally, given a vector $\mathbf{X} = (X_1, ..., X_m)$ of variables, let $\mathbf{X}^\mathbf{i}$ denote the monomial of degree $\mathrm{wt}\,\mathbf{i}$ given by the following expression,

$$\mathbf{X}^\mathbf{i} := \prod_j X_j^{i_j} \in \mathbb{F}_q[X_1, ..., X_m] = \mathbb{F}_q[\mathbf{X}].$$

To formalize this idea of using derivative of polynomials, we have the following definition:

Definition 4. Given a polynomial $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ and a vector of non-negative integers $\mathbf{i}$, the $\mathbf{i}$th **Hasse Derivative** is the polynomial coefficient of $\mathbf{Z}^{\mathbf{i}}$, denoted $P^{(\mathbf{i})}(\mathbf{X})$, from the expansion

$$P(\mathbf{X}+\mathbf{Z}) = \sum_{\mathbf{i}} P^{(\mathbf{i})}(\mathbf{X})\mathbf{Z}^{\mathbf{i}}.$$

The Hasse derivative aligns with the standard definition of partial derivatives, up to a constant factor based on $\mathbf{i}$. However, this definition is more useful for defining and proving facts about multiplicity codes, as well as being nicely defined over finite fields with potentially small character.

Definition 5. Given $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}], \mathbf{a} \in \mathbb{F}_q^m$, and $s \in \mathbb{Z}_{\geq 0}$, the **order** s **evaluation** of P at $\mathbf{a}$ is

$$P^{(<s)}(\mathbf{a}) = \langle P^{(\mathbf{i})}(\mathbf{a}) : \operatorname{wt}\mathbf{i} < s\rangle \in \mathbb{F}_q^{|\{\operatorname{wt}\mathbf{i}<s\}|} = \mathbb{F}_q^{\binom{m+s-1}{m}}$$

Now, we have the tools to formally define multiplicity codes:

Definition 6. The (s,d,m,q)**-multiplicity code** is defined over the alphabet $\Sigma = \mathbb{F}_q^{\binom{m+s-1}{m}}$, and of length $n = q^m$ over this alphabet. Define an encoding map

$$\mathrm{Enc}_{s,d,m,q} : \mathbb{F}_q[\mathbf{X}] \to \Sigma^n, \qquad P \mapsto \langle P^{(<s)}(\mathbf{a}) : \mathbf{a} \in \mathbb{F}_q^m\rangle.$$

Given this, the (s,d,m,q)-multiplicity code is $\mathcal{C} = \{\mathrm{Enc}_{s,d,m,q}(P) : \deg P \leq d\}$

For example, Reed Muller codes as multiplicity codes with $s = 1$ and $\mathcal{C}_{toy}$ is a $(2,d,2,q)$-multiplicity code. Now, we move on to define various features of multiplicity codes. To do so, we need one more definition.

Definition 7. Given $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$, and an evaluation point $\mathbf{a} \in \mathbb{F}_q^m$, the **multiplicity** of P at $\mathbf{a}$ is

$$\mathrm{mult}(P,\mathbf{a}) := \max\{M \in \mathbb{Z}_{\geq 0} : P^{(\mathbf{i})}(\mathbf{a}) = 0, \quad \forall \mathbf{i} \text{ s.t. } \operatorname{wt}\mathbf{i} < M\}.$$

This notion of multiplicity agrees with the usual definition of univariate multiplicity: given $f(x) \in \mathbb{R}[x]$, $a \in \mathbb{R}$ has multiplicity M if $f^{(k)}(a) = 0$ for all $0 \leq k < M$. The definition above extends the common notion of multiplicity to multivariate polynomials, and accounts for the Hasse derivative. This allows us to generalize the statement "low degree polynomials don't have too many roots."

Lemma 2.1. *Let $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ be a polynomial of total degree d. Then for any $S \subset \mathbb{F}_q$,*

$$\sum_{\mathbf{a}\in S^n} \mathrm{mult}(P,\mathbf{a}) \leq d|S|^{n-1}$$

In particular, for any $s > 0$,

$$\Pr_{\mathbf{a}\in S^n}[\mathrm{mult}(P,\mathbf{a}) \geq s] \leq \frac{d}{s|S|}$$

This will be taken as a fact: the proof is analogous to the univariate case. It is proven elsewhere, for example in [1]. So, what does this tell us about multiplicity codes? This fact about roots of low degree polynomials allows us to prove basic properties of multiplicity codes.

Theorem 2.2. *Let $\mathcal{C}$ be a (s,d,m,q)-multiplicity code. Then $\mathcal{C}$ has distance*

$$\delta = 1 - \frac{d}{sq},$$

and rate R where

$$R = \frac{\binom{d+m}{m}}{\binom{m+s-1}{m} q^m} \geq \left(1 - \frac{m^2}{s}\right)(1-\delta)^m.$$

Remark. We can see that from this lower bound of R that we can choose m, s, d, q such that the rate gets arbitrarily close to 1. This is an advantage of multiplicity codes over Reed Muller codes and other local correction codes.

Proof. To compute the distance, consider any two codewords c_1, c_2, corresponding to two polynomials $P_1, P_2 \in \mathbb{F}_q[\mathbf{X}]$. If they agree at coordinate $\mathbf{a} \in \mathbb{F}_q^m$, then $P_1^{(<s)}(\mathbf{a}) = P_2^{(<s)}(\mathbf{a})$, so $(P_1 - P_2)^{(<s)}(\mathbf{a}) = 0$. Hence, $\text{mult}(P_1 - P_2, \mathbf{a}) \geq s$. As stated in Lemma 2.1, for $S = \mathbb{F}_q$, this can happen for at most $\frac{d}{sq}$ fraction of the $a \in \mathbb{F}_q^m$. As such, the distance is at least $1 - \frac{d}{sq}$. It is easy to verify that this is tight, so $\delta = 1 - \frac{d}{sq}$.

On the other hand, the rate is easy to prove: there are $\binom{d+m}{m}$ coefficients on m-variate polynomials of degree at most d, so the space of multivariate polynomials of degree at most d has dimension $\binom{d+m}{m}$. So, over $\mathbb{F}_q$, the size of the input space is $q^{\binom{d+m}{m}}$. Every polynomial gives a unique decoding, hence $k = \binom{d+m}{m}$. For n, we see that each symbol in the output alphabet is in $\mathbb{F}_q^{\binom{m+s-1}{m}}$, and there are $n = q^m$ of them. Thus the rate is as stated above.

To verify the stated bound, note that

$$\begin{aligned} R = \frac{\binom{d+m}{m}}{\binom{m+s-1}{m} q^m} &= \prod_{j=0}^{m-1} \frac{d+m-j}{(s+m-j-1)q} \\ &\geq \left(\frac{d}{s+m}\right)^m \left(\frac{1}{q}\right)^m \\ &= \left(\frac{1}{1+\frac{m}{s}}\right)^m \left(\frac{d}{sq}\right)^m \\ &\geq \left(1 - \frac{m^2}{s}\right)(1-\delta)^m. \end{aligned}$$

□

3. Decoding scheme

We saw that Berlekamp-Welch algorithm decodes Reed Solomon codes efficiently. We will present an extension of this algorithm for univariate multiplicity codes described by Kopparty in [2]. This algorithm will gives efficient unique decoding. Then we will reduce the problem of list decoding multivariate multiplicity codes to the case of univariate multiplicity codes.

3.1. **Unique decoding for univariate multiplicity codes.** We will present the algorithm for unique decoding univariate multiplicity codes $\mathrm{Enc}_{s,d,1,q}(P)$.

Output: $P(X)$ such that $\delta(\mathrm{Enc}_{s,d,1,q}(P), r) < \delta/2$.

1. Find polynomials $E(X)$, $N(X)$ of degree at most $(sq-d)/2$ and $(sq+d)/2$ respectively such that
$$\begin{aligned} N(X) &= E(X)r^{(0)}(X) \\ N^{(1)}(X) &= E(X)r^{(1)}(X) + E^{(1)}(X)r^{(0)}(X) \\ &\vdots \\ N^{(s-1)}(X) &= \sum_{i=0}^{s-1} E^{(i)}(X)r^{(s-1-i)}(X) \end{aligned}$$
2. Output $N(X)/E(X)$.

Now we analyze the correctness and runtime of this algorithm. In terms of correctness, we want to show that if there exists $P(X)$ satisfying $\delta(\mathrm{Enc}_{s,d,1,q}(P), r) < \delta/2$ then the algorithm produces $P(X)$.

First we need to show that step (1) outputs a $N(X), E(X)$. We note this is sq linear equations with $(sq+d)/2 + 1 + (sq-d)/2 > sq$ unknowns for the coefficients of E and N. So, a nonzero solution of $N(X)$ and $E(X)$ exists. Next, we note that if such a $P(X)$ of degree $\leq d$ exits, $N(X) - P(X)E(X)$ is a polynomial of degree $(sq+d)/2$. Then let us count the zeros of $N(X) - P(X)E(X)$ with multiplicity at least s. Since r and $\mathrm{Enc}_{s,d,1,q}(P)$ differ at less than $\delta n/2$ spots, $N(X) - P(X)E(X)$ has zeros at $> (sd+q)/(2s)$ spots with multiplicity at least s. Hence, $N(X) - P(X)E(X) = 0$ and the algorithm indeed outputs $P(X)$.

We will show that this algorithm runs in near linear time. We break down the process for finding $N(X)$ and $E(X)$.

1. Find $R(X)$ of degree at most $sq-1$ such that
$$R^{(<s)}(\alpha) = r^{(<s)}(\alpha)$$
for each $\alpha \in \mathbb{F}_q$. This can be done by Hermite interpolation in near linear time, which is similar to Newton's polynomial interpolation but takes into fact the derivatives of a polynomial.
2. Find $C(X), E(X)$ such that $\deg(E(X)) \leq (sq-d)/2$ and $C(X)/E(X)$ approximates $R(X)/(X^q - X)^s$ where the difference of numerators over a common denominator has degree at most $(sq+d)/2$. This can be accomplished using Strassen's continued fraction expansion as described in [5].
3. Output $E(X)$ and $N(X) = E(X)R(X) - C(X)(X^q - X)^s$.

Both Hermite interpolation and Strassen's continued fraction algorithm can be completed in near linear time as stated in [2] and thus we have a nearly linear unique decoding of univariate multiplicity codes.

3.2. **List decoding.** In addition to the unique decoding as defined above, multiplicity codes allow for list decoding. List decoding over univariate multiplicity codes is a generalization of the Guruswami-Sudan algorithm, and allows for list decoding from a fraction of errors up to the Johnson bound. This algorithm is discussed in detail in [3], but is far too complicated to cover in this summary. As we are interested in general multivariate multiplicity codes, we will show how the multivariate case reduces to the univariate one. In order to do so, we use a few definitions and lemmas:

Definition 8. An element $\mathbf{a} \in \mathbb{F}_{q^m}^m$ is a **basis** if its coordinates form a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. For a basis $\mathbf{a} = (a_1, .., a_m)$, we associate a curve

$$\gamma_{\mathbf{a}}(T) = (\mathrm{Tr}(a_1 T), ..., \mathrm{Tr}(a_m T)) \in \mathbb{F}_q[T]^m$$

where Tr notates the trace map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$:

$$\mathrm{Tr}(t) = t + t^q + ... + t^{q^{m-1}}.$$

It turns out that this is a bijection between $\mathbb{F}_{q^m}$ and $\mathbb{F}_q^m$.

An important property of $\gamma_{\mathbf{a}}$ is that it allows us to relate evaluations of multivariate polynomials and their derivatives over $\mathbb{F}_q^m$ to univariate polynomials over $\mathbb{F}_{q^m}$.

Lemma 3.1. *Let $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$, and $Q(T) = P \circ \gamma_{\mathbf{a}}(T)$. Then for all $t \in \mathbb{F}_{q^m}, j < q$,*

$$Q^{(j)}(t) = \sum_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j} P^{(j)}(\gamma_{\mathbf{a}}(t))\mathbf{a}^{\mathbf{i}}$$

This lemma follows from evaluating $Q^{(j)}(T)$ and using the definition of Hasse derivatives. We refer the reader to [3] for the explicit computation. Next we introduce the definition of s-general position. An intuitive definition of s-general position is that if $R(\mathbf{X})$ has degree at most s and has a zero at all $\mathbf{a}_1, ..., \mathbf{a}_M$ then R is zero. More formally, we have the following.

Definition 9. A collection of bases $\mathbf{a}_1, ..., \mathbf{a}_M$ is in $s-$ **general position** if for all non-zero $R(\mathbf{X}) \in \mathbb{F}_{q^m}[\mathbf{X}]$ of degree at most s, there is some $e \in [M]$ such that $R(\mathbf{a}_e) \neq 0$.

Lemma 3.2. *Let $s < q$. Let $Q(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ have degree $d < sq$. Let $\mathbf{a}_1, ..., \mathbf{a}_M$ be bases for $\mathbb{F}_{q^m}$ in s-general position. If for each $i \in [M]$, $Q \circ \gamma_{\mathbf{a}_i}(T) = 0$, then $Q(\mathbf{X}) = 0$.*

Proof. We will show that if $Q \circ \gamma_{\mathbf{a}_i}(T) = 0$ for each $\mathbf{a}_i$, then $\mathrm{mult}(Q, \mathbf{b}) \geq s$ for each $\mathbf{b} \in \mathbb{F}_q^m$. Then by Lemma 2.1, $Q = 0$. Let $e \in [M]$. Then Lemma 3.1 gives us that

$$\sum_{\mathbf{i}|\mathrm{wt}\,\mathbf{i}=j} Q^{(\mathbf{i})} \circ \gamma_{\mathbf{a}_e}(T)\mathbf{a}_e^{\mathbf{i}} = 0.$$

Since we range over $t \in \mathbb{F}_{q^m}$ and $\gamma_{\mathbf{a}_e}$ gives a bijection, this shows that for all $\mathbf{b} \in \mathbb{F}_q^m$:

$$\sum_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j} Q^{(\mathbf{i})}(\mathbf{b})\mathbf{a}_e^{\mathbf{i}} = 0.$$

Then if we define $R_{\mathbf{b},j}(\mathbf{Y}) := \sum_{\mathbf{i}:\mathrm{wt}\,i=j} Q^{(\mathbf{i})}(\mathbf{b})\mathbf{Y}^{\mathbf{i}}$. Then for $j < s$, since $R_{\mathbf{b},j}(\mathbf{a}_e) = 0$ for each $e \in [M]$, we have that $R_{\mathbf{b},j} = 0$ since $\mathbf{a}_1, ..., \mathbf{a}_M$ is in s-general position. Looking at the definition of $R_{\mathbf{b},j}$ this means that $Q^{(\mathbf{i})}(\mathbf{b}) = 0$ for each $\mathbf{b} \in \mathbb{F}_q^m$ and $\mathrm{wt}\,\mathbf{i} < s$. So, $\mathrm{mult}(Q, \mathbf{b}) \geq s$ for each $\mathbf{b} \in \mathbb{F}_q^m$ and thus $Q = 0$. □

Given this setup, we can now define an algorithm to reduce multivariate-decoding to the univariate case. This algorithm was presented by Kopparty in [3].

Setup: An algorithm $\mathcal{A}$ that can list-decode univariate multiplicity codes of distance δ, from $\eta(\delta)$ fraction of errors. (In §3.1 we describe a decoding algorithm, and in [2] a specific list decoding algorithm for univariate codes is given). Let us consider a (s, d, m, q) a multiplicity code, with distance δ_0.

Input: A received message $r \in \Sigma^n$.

(1) Let $M = \binom{m+s-1}{m}$, and pick bases $\mathbf{a}_1, ..., \mathbf{a}_M \in \mathbb{F}_{q^m}^m$ in s-general position.

(2) For $i \in [M]$, define $\ell_i : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}^s$ by ranging $0 \le j < s$ and writing the jth component as

$$(\ell_i(t))_j := \sum_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j} r^{(\mathbf{i})}(\gamma_{\mathbf{a}_i}(t))\mathbf{a}_i^{\mathbf{i}}$$

(3) Using $\mathcal{A}$, list-decode ℓ_i, i.e. compute $\mathcal{L}_i$ such that for $Q(T) \in \mathcal{L}_i, \deg Q \le dq^{m-1}$,

$$\delta(\mathrm{Enc}_{s,dq^{m-1},1,q^m}(Q), \ell_i) < \delta_0$$

(4) For all $(Q_1, ..., Q_M) \in \prod_{i=1}^M \mathcal{L}_i$, find all $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ such that $\deg P \le d$, and for all $i \in [M]$, $P \circ \gamma_{\mathbf{a}_i}(T) = Q_i(T)$.

(5) Return a list of all such $P(\mathbf{X})$.

To show that this algorithm is correct, consider $P(\mathbf{X})$ such that $\delta(\mathrm{Enc}_{s,d,m,q}(P), r) < \delta_0$. Let $E \subset \mathbb{F}_q^m$ be the set of indices $\mathbf{a}$ such that $P^{(<s)}(\mathbf{a}) \ne r^{(<s)}(\mathbf{a})$. Define $Q_i(T) = P \circ \gamma_{\mathbf{a}_i}(T) \in \mathcal{L}_i$. By the second lemma above, for every t such that $\gamma_{\mathbf{a}}(t) \notin E, j < s$,

$$Q_i^{(j)}(t) = \sum_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j} P^{(j)}(\gamma_{\mathbf{a}}(t))\mathbf{a}^{\mathbf{i}} = \sum_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j} r^{(j)}(\gamma_{\mathbf{a}}(t))\mathbf{a}^{\mathbf{i}}$$

This means that $\delta(\mathrm{Enc}_{s-c+1,d,1,q}(Q_i), \ell_{e,\mathbf{i}}) \le \frac{|E|}{q^m} < \delta_0$, so $Q_i \in \mathcal{L}_i$. Thus, $\mathcal{A}$ will output Q_i when finding $\mathcal{L}_{e,\mathbf{i}}$. Therefore, this algorithm allows us to reduce the general m-variate case to the univariate case.

To discuss briefly the efficiency of this algorithm, we note that Lemma 3.1 allows us to state that for any given $(Q_1, ..., Q_M)$ vector, there is a unique solution to step (5). If both $P_1, P_2 \in \mathbb{F}_q[\mathbf{X}]$ satisfy the desired system of equations, $(P_1 - P_2) \circ \gamma_{\mathbf{a}_e} = 0$ for each e. So $P_1 = P_2$. Thus, as long as we can compute P efficiently in step (5), this algorithm is efficient. However, we have a system of equations with $\binom{d+m}{m}$ variables and $(d+1)M$ constraints, so we can find solutions efficiently. This shows that multiplicity codes are efficiently list decodable as stated in [2].

4. Local correction

We saw in §1.1.1 a local correction algorithm for the $(2, d, 2, q)$-multiplicity codes $\mathcal{C}_{toy}$. We will expand on that algorithm to locally correct multiplicity codes in general. In particular, we will prove the following theorem.

Theorem 4.1. *Let $\mathcal{C}$ be a (s, d, m, q)-multiplicity code. Let $\delta = 1 - \frac{d}{sq}$ be the distance of $\mathcal{C}$. Suppose $q \ge \max(10m, \frac{d+6}{s}, 5(s+1))$. Then $\mathcal{C}$ is locally correctable from $\frac{\delta}{10}$-fraction of errors with $q \cdot O(s^m)$-queries with runtime poly$(s^m, q^{O(1)})$.*

4.1. Local correction algorithm for (s, d, m, q)-multiplicity codes.

The following algorithm and analysis will give the proof of the theorem. This algorithm was presented in [2] and [4].

Input: $r \in \Sigma^n$, $\mathbf{a} \in \mathbb{F}_q^m$.

Output: $(P^{(<s)}(\mathbf{a}))$.

(1) **Pick directions**: Pick $\mathbf{z}, \mathbf{y}_1, ..., \mathbf{y}_m \overset{R}{\leftarrow} \mathbb{F}_q^m$. Let $S \subset \mathbb{F}_q$ be a subset of size $5(s+1)$. Define the set of directions

$$B = \{\mathbf{z} + \sum_{i=1}^{m} \alpha_i \mathbf{y}_i \in \mathbb{F}_q^m : \alpha_i \in S\}$$

(2) **Solve for $Q_{\mathbf{b}}(T)$ for each $\mathbf{b} \in B$**: Let us denote $r_{(\mathbf{i})}(\mathbf{a})$ as the **i**th coordinate of r at position $\mathbf{a}$. We will define the function $\ell_{\mathbf{b}} : \mathbb{F}_q \to \mathbb{F}_q^s$. First we define each coordinate:

$$\ell_{\mathbf{b}}(T)_j := \sum_{\mathbf{i} | \mathrm{wt}(\mathbf{i}) = j} r_{(\mathbf{i})}(\mathbf{a} + \mathbf{b}T)\mathbf{b}^{\mathbf{i}}$$

Combining these together, we have the function:

$$\ell_{\mathbf{b}}(T) := (\ell_{\mathbf{b}}(T)_0, \ell_{\mathbf{b}}(T)_1, ..., \ell_{\mathbf{b}}(T)_{s-1})$$

Using the unique decoding algorithm for univariate polynomials described in §3.1, we find $Q_{\mathbf{b}}(T)$ of degree $\leq d$ (if such a polynomial exists) where $\delta(\mathrm{Enc}_{s,d,1,q}(Q_{\mathbf{b}}), \ell_{\mathbf{b}}) < \frac{\delta}{2}$.

(3) **Solve a noisy linear system**: For each $0 \leq j < s$, consider the system in variables $u_{\mathbf{i}}$ where $\mathrm{wt}(\mathbf{i}) = j$. For each $\mathbf{b} \in B$, we get an equation:

$$\sum_{\mathbf{i} | \mathrm{wt}(\mathbf{i}) = j} \mathbf{b}^{\mathbf{i}} u_{\mathbf{i}} = \text{coeff of } T^j \text{ in } Q_{\mathbf{b}}(T)$$

Find all $(u_{\mathbf{i}})_{\mathbf{i}:\mathrm{wt}\,\mathbf{i}=j}$ satisfying the equations for at least 3/5 of the equations given by various $\mathbf{b} \in B$. If there are 0 or > 1 satisfying the equation, output FAIL.

(4) Output $(u_{\mathbf{i}})_{\mathrm{wt}(\mathbf{i})<s}$ in the order of $\mathbf{i}$ used for the encoding scheme.

Now we want to show that this algorithm corrects for up to $\delta/10$ fraction of errors. Assume $P(\mathbf{X})$ is the polynomial such that $\delta(r, \mathrm{Enc}_{s,d,m,q}(P)) < \delta/10$. We want to show that we indeed recover $(P^{(<s)}(\mathbf{a}))$ with the desired queries and running time.

4.1.1. *Recovering* $(P^{(<s)}(\mathbf{a}))$. We will split this part of the proof into three main steps.

1. Many of the directions b have few errors: First we consider the set $\mathbb{F}_q^m/\{0\}$ and generally the number of errors on the line $L_{\mathbf{d}} = \{\mathbf{a} + \mathbf{d}t \mid t \in \mathbb{F}_q\}$ for $\mathbf{d} \in \mathbb{F}_q^m/\{0\}$. These lines cover $\mathbb{F}_q^m/\{\mathbf{a}\}$ uniformly as we vary $\mathbf{d}$; hence, at least 2/3 of the lines satisfy that there are fewer than $(\delta/3 + 1/q) < \frac{\delta}{2}$ fraction of errors on them (note that this inequality comes from the fact that $q > \frac{d+6}{s} \geq \delta/6$). Such a direction with fewer than $\frac{\delta}{2}$ fraction of errors we will call a *low error direction*. Now we want to see how many of these low error directions B covers.

We note that the set of low error directions $E = \{\mathbf{d} \in \mathbb{F}_q^m/\{0\} \mid \mathbf{d} \text{ is a low error direction}\}$ has the property that $|E| \geq 2q^m/3$ from the argument above. Now, since $\mathbf{z}, \mathbf{y}_1, ..., \mathbf{y}_m$ are chosen randomly and independently from $\mathbb{F}_q^m$, B is a collection of pairwise independent random variables $\{\mathbf{v}_{\alpha_1,...,\alpha_m} = \mathbf{z} + \sum_{i=1}^m \alpha_i \mathbf{y}_i\}$. Then consider the expression

$$\Pr\left(\sum_{\alpha_1,...,\alpha_m \in S^m} \mathbf{1}[\mathbf{v}_{\alpha_1,...,\alpha_m} \in E] \leq \frac{3|S|^m}{5}\right)$$

Applying Chebyshev's inequality, we see that this is < 0.1. So, with probability at least 0.9, at least 3/5 of B are low error direction.

2. If $\mathbf{b}$ is a low error direction, $Q_{\mathbf{b}}(T) = P(\mathbf{a} + \mathbf{b}T)$: Consider $Q(T) = P(\mathbf{a} + \mathbf{b}T)$. Then a simple expansion of equations gives us that

$$Q^{(j)}(T) = \sum_{\mathbf{i}|\mathrm{wt}(\mathbf{i})=j} P^{(\mathbf{i})}(\mathbf{a} + \mathbf{b}T)\mathbf{b}^{\mathbf{i}}$$

So for $\mathbf{b}$ a lower error direction, $\delta(\mathrm{Enc}_{s,d,1,q}(Q), \ell_{\mathbf{b}}) < \delta/2$. Then this implies that $Q(T) = Q_{\mathbf{b}}(T)$ since there can not be more than codeword satisfying this property.

3. If $\mathbf{b}$ is a low error direction, $u_{\mathbf{i}} = P^{(\mathbf{i})}(\mathbf{a})$ **for each $\mathbf{i}$**: Since $Q_{\mathbf{b}}(T) = P(\mathbf{a} + \mathbf{b}T)$ for a low error direction, we can instead analyze $Q(T) = P(\mathbf{a} + \mathbf{b}T)$. Using the definition of Hasse derivatives,

$$Q(T) = \sum_{\mathbf{i}|\mathrm{wt}(i)} P^{(\mathbf{i})}(\mathbf{a})\mathbf{b}^{i}T^{\mathrm{wt}(i)}.$$

So if we group the terms,

$$\sum_{\mathbf{i}|\mathrm{wt}(\mathbf{i})=j} P^{(\mathbf{i})}(\mathbf{a})\mathbf{b}^{i} = \text{coeff of } T^{j} \text{ in } Q(T).$$

So, $P^{(\mathbf{i})}(\mathbf{a})$ should satisfy the at least 3/5 of the linear system of equations. Finally, we need to show that this solution of $u_{\mathbf{i}}$ is unique with high probability. Let us look at $\mathbf{i}$ with weight j. With probability at least 0.9, the $\mathbf{y}_1, ..., \mathbf{y}_m$ will be linearly independent over $\mathbb{F}_q^m$. So, we have a linear bijection taking $S^m \tilde{\rightarrow} B$. Then we claim for now that there is no polynomial of degree $< s$ vanishing on more than 1/5 of S^m. This tells us there are no polynomials of degree $< s$ vanishing on more than 1/5 of B.

Assuming this claim, we show that $(u_{\mathbf{i}})_{\mathbf{i}}$ is unique if it satisfies at least 3/5 of the equations. Assume $(u_{\mathbf{i}})_{\mathbf{i}}$ and $(u_{\mathbf{i}})'_{\mathbf{i}}$ both satisfy at least 3/5 of the equations. Then they must satisfy at least 1/5 of the same equations. So, $(u_{\mathbf{i}} - u'_{\mathbf{i}})_{\mathbf{i}}$ give the coefficients on a degree $< s$ polynomial that vanishes on at least 1/5 of B. Hence, we have a contradiction.

Finally, we need to prove the claim. This follows from Lemma 2.1 using degree s and recalling that $|S| = 5(s+1)$. Thus, we have proven that the algorithm outputs $P^{(<s)}(\mathbf{a})$.

4.1.2. *Query complexity.* We can see that for each $\mathbf{b} \in B$, we query for each $T \in \mathbb{F}_q$. Then $|B| = O(s^m)$, so we have $q \cdot O(s^m)$ queries. We note that this is a small number in terms of n.

4.1.3. *Runtime.* The main two algorithms that we need to discuss the runtime of are decoding the univariate multiplicity code and solving the noisy system of equations. We saw in §4.1 that the univariate multiplicity code can be decoded in near linear time. For the noisy system of equations, using the bijection from $S^m \to B$, we are reduced to solving a problem of noisy polynomial interpolation on S^m. In [4], Kopparty, Saraf and Yekhanin describe how to do this in $\mathrm{poly}(|S|^m, \log(q))$ for $S = \mathbb{F}_p \subset \mathbb{F}_q$. Since $|S| = 5(s+1)$, we have our desired result.

Proof of Theorem 1.1. We pick $m = [1/\epsilon]$. For large enough q, we construct the code with $n = q^m$. Then observe that $(1-\alpha) < (1-\delta)^m$ since

$$(1-\alpha) < (1-\epsilon\alpha/2)^{2/\epsilon} < (1-\delta)^m$$

Let s satisfy:

$$1 - \frac{m^2}{s} > \frac{1-\alpha}{(1-\delta)^m}$$

Let $d = (1-\delta)sq$. Then we consider the (s, d, m, q) multiplicity code has block length n and alphabet over $n^{O(1)}$. Theorem 2.2 tells that this code has distance δ and rate

$$R = \frac{\binom{d+m}{m}}{\binom{m+s-1}{m} q^m} \geq (1 - \frac{m^2}{s})(1-\delta)^m > 1 - \alpha$$

Theorem 4.1 tells us that this corrects $\delta/10$ fraction of errors with $O(n^\epsilon)$ queries. Thus, this code gives us the desired bound for Theorem 1.1

□

5. Encoding scheme

When defining multiplicity codes, we used a specific encoding map to transform polynomials in $\mathbb{F}_q[\mathbf{X}]$ into codewords, which lie in $\mathcal{C} \subset \Sigma^{q^m}$. Note that the multiplicity codes always form a $\mathbb{F}_q$-linear subspace of Σ^{q^m}, and so we would like to extend this encoding to be a $\mathbb{F}_q$-linear encoding as well. The natural way is to look at the map

$$E : \mathbb{F}_q^{\binom{d+m}{d}} \to \mathcal{C}, \qquad c \mapsto E_{s,d,m,q}(c(\mathbf{X})),$$

where $c(\mathbf{X})$ is the polynomial with coefficients that are entries of c. Given any vector c, computing this polynomial $P = c(\mathbf{X})$, as well as its Hasse derivatives, can be done in near-linear time (specifically, $O((d^m + q^m)\binom{m+s}{m}\log(d^m + q^m)))$.

However, it turns out that there is another problem that multiplicity codes can solve, known as local decoding:

Definition 10. A code $\mathcal{C}$ with encoding $E : \Sigma_0^k \to \mathcal{C} \subset \Sigma^n$ is **locally decodable** from η fraction of errors if given some $i \in [k]$ and $r \in \Sigma^n$, one can recover x_i from the unique $x \in \Sigma_0^k$ such that $\delta(E(x), r) < \eta$.

The query complexity and runtime are the same as for locally correctable codes.

In order to locally decode, we desire a systematic encoding, i.e. where the message symbols appear in the codeword. Given such a map, the local correction algorithm described will immediately give rise to a local decoding algorithm, because querying a bit of the encoded message in Σ_0^k simply requires knowing the corresponding bit of the codeword. To create such an encoding, we make another definition.

Definition 11. For a fixed (s, d, m, q), an **interpolating set** is a set $S \subset \mathbb{F}_q^m \times \{\mathbf{i} : \operatorname{wt} \mathbf{i} < s\}$ such that for every $f : S \to \mathbb{F}_q$, there is a unique $P(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ of degree at most d such that for all $(\mathbf{a}, \mathbf{i}) \in S, P^{(\mathbf{i})}(\mathbf{a}) = f(\mathbf{a}, \mathbf{i})$.

It is relatively clear that such S exist, and therefore have size $\binom{d+m}{d}$. However, for the local decoding algorithm to run efficiently, the set S needs to be explicit. While this is not obvious, it is shown in [3] that one can construct explicit interpolating sets for any (s, d, m, q). In fact, these interpolating sets are given by combinations of interpolating sets for the $s = 1$ case, i.e. over Reed-Muller codes. Assuming their existence, this allows the local correction algorithm to give rise to an efficient local decoding algorithm.

6. Conclusion

The multiplicity codes introduced here build on the local correctability of Reed-Muller codes. However, these codes also provide a greater range of possible rates and query sizes as discussed in Theorem 1.1. In particular, we have seen that multiplicity codes can be constructed to have arbitrarily high rate, while simultaneously having an reasonably low query size. While this query size is not as low as the constant or logarithmic query size that some Reed Muller codes can achieve, it is still a good tradeoff of rate and query complexity. These high rate codes are of particular interest for storage. With these codes, one can store data more efficiently than using other locally correctable codes (due to the high rate), while providing good error-proofing.

There are still a number of open questions surrounding these codes. As outlined in the global decoding section, there is a list decoding algorithm for arbitrary multiplicity codes that reduces to an algorithm for univariate codes. However, it is still unknown what the list decoding radius for univariate codes is. It has been shown for a fraction of errors $\eta \leq 1 - \sqrt{1-\delta}$ for arbitrary codes, while for prime fields $\mathbb{F}_q$, and large enough s, radius $\eta = \delta$ has been proven. For more open questions around multiplicity codes, we refer the reader to section 6 and 7 of [2].

Although there are many unanswered questions about multiplicity codes, it is possible that they will become as important to the field of coding theory as their related counterparts Reed-Solomon and Reed-Muller codes. Because of their high rate and decent query complexity and ability to both locally correct and locally decode, multiplicity codes have high potential for future applications.

References

[1] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan. Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. *SIAM J. Comput.*, 42(6):2305–2328, 2013.

[2] S. Kopparty. Some remarks on multiplicity codes. In *Discrete geometry and algebraic combinatorics*, volume 625 of *Contemp. Math.*, pages 155–176. Amer. Math. Soc., Providence, RI, 2014.

[3] S. Kopparty. List-decoding multiplicity codes. *Theory Comput.*, 11:149–182, 2015.

[4] S. Kopparty, S. Saraf, and S. Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):Art. 28, 20, 2014.

[5] V. Strassen. The computational complexity of continued fractions. *SIAM J. Comput.*, 12(1):1–27, 1983.